NI Vision

NI Vision for LabVIEW™ User Manual

November 2005
371007B-01

NATIONAL INSTRUMENTS™

Worldwide Technical Support and Product Information

ni.com

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759-3504 USA Tel: 512 683 0100

Worldwide Offices

Australia 1800 300 800, Austria 43 0 662 45 79 90 0, Belgium 32 0 2 757 00 20, Brazil 55 11 3262 3599, Canada 800 433 3488, China 86 21 6555 7838, Czech Republic 420 224 235 774, Denmark 45 45 76 26 00, Finland 385 0 9 725 725 11, France 33 0 1 48 14 24 24, Germany 49 0 89 741 31 30, India 91 80 51190000, Israel 972 0 3 6393737, Italy 39 02 413091, Japan 81 3 5472 2970, Korea 82 02 3451 3400, Lebanon 961 0 1 33 28 28, Malaysia 1800 887710, Mexico 01 800 010 0793, Netherlands 31 0 348 433 466, New Zealand 0800 553 322, Norway 47 0 66 90 76 60, Poland 48 22 3390150, Portugal 351 210 311 210, Russia 7 095 783 68 51, Singapore 1800 226 5886, Slovenia 386 3 425 4200, South Africa 27 0 11 805 8197, Spain 34 91 640 0085, Sweden 46 0 8 587 895 00, Switzerland 41 56 200 51 51, Taiwan 886 02 2377 2222, Thailand 662 278 6777, United Kingdom 44 0 1635 523545

For further support information, refer to the *Technical Support and Professional Services* appendix. To comment on National Instruments documentation, refer to the National Instruments Web site at ni.com/info and enter the info code feedback.

© 2000–2005 National Instruments Corporation. All rights reserved.

Important Information

Warranty

The media is warranted against defects in materials and workmanship for a period of 90 days from the date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace equipment that proves to be defective during the warranty period. This warranty includes parts and labor.

The media on which you receive National Instruments software are warranted not to fail to execute programming instructions, due to defects in materials and workmanship, for a period of 90 days from date of shipment, as evidenced by receipts or other documentation. National Instruments will, at its option, repair or replace software media that do not execute programming instructions if National Instruments receives notice of such defects during the warranty period. National Instruments does not warrant that the operation of the software shall be uninterrupted or error free.

A Return Material Authorization (RMA) number must be obtained from the factory and clearly marked on the outside of the package before any equipment will be accepted for warranty work. National Instruments will pay the shipping costs of returning to the owner parts which are covered by warranty.

National Instruments believes that the information in this document is accurate. The document has been carefully reviewed for technical accuracy. In the event that technical or typographical errors exist, National Instruments reserves the right to make changes to subsequent editions of this document without prior notice to holders of this edition. The reader should consult National Instruments if errors are suspected. In no event shall National Instruments be liable for any damages arising out of or related to this document or the information contained in it.

EXCEPT AS SPECIFIED HEREIN, NATIONAL INSTRUMENTS MAKES NO WARRANTIES, EXPRESS OR IMPLIED, AND SPECIFICALLY DISCLAIMS ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. CUSTOMER'S RIGHT TO RECOVER DAMAGES CAUSED BY FAULT OR NEGLIGENCE ON THE PART OF NATIONAL INSTRUMENTS SHALL BE LIMITED TO THE AMOUNT THERETOFORE PAID BY THE CUSTOMER. NATIONAL INSTRUMENTS WILL NOT BE LIABLE FOR DAMAGES RESULTING FROM LOSS OF DATA, PROFITS, USE OF PRODUCTS, OR INCIDENTAL OR CONSEQUENTIAL DAMAGES, EVEN IF ADVISED OF THE POSSIBILITY THEREOF. This limitation of the liability of National Instruments will apply regardless of the form of action, whether in contract or tort, including negligence. Any action against National Instruments must be brought within one year after the cause of action accrues. National Instruments shall not be liable for any delay in performance due to causes beyond its reasonable control. The warranty provided herein does not cover damages, defects, malfunctions, or service failures caused by owner's failure to follow the National Instruments installation, operation, or maintenance instructions; owner's modification of the product; owner's abuse, misuse, or negligent acts; and power failure or surges, fire, flood, accident, actions of third parties, or other events outside reasonable control.

Copyright

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

Trademarks

National Instruments, NI, ni.com, and LabVIEW are trademarks of National Instruments Corporation. Refer to the *Terms of Use* section on `ni.com/legal` for more information about National Instruments trademarks.

Other product and company names mentioned herein are trademarks or trade names of their respective companies.

Members of the National Instruments Alliance Partner Program are business entities independent from National Instruments and have no agency, partnership, or joint-venture relationship with National Instruments.

Patents

For patents covering National Instruments products, refer to the appropriate location: **Help»Patents** in your software, the `patents.txt` file on your CD, or `ni.com/patents`.

WARNING REGARDING USE OF NATIONAL INSTRUMENTS PRODUCTS

(1) NATIONAL INSTRUMENTS PRODUCTS ARE NOT DESIGNED WITH COMPONENTS AND TESTING FOR A LEVEL OF RELIABILITY SUITABLE FOR USE IN OR IN CONNECTION WITH SURGICAL IMPLANTS OR AS CRITICAL COMPONENTS IN ANY LIFE SUPPORT SYSTEMS WHOSE FAILURE TO PERFORM CAN REASONABLY BE EXPECTED TO CAUSE SIGNIFICANT INJURY TO A HUMAN.

(2) IN ANY APPLICATION, INCLUDING THE ABOVE, RELIABILITY OF OPERATION OF THE SOFTWARE PRODUCTS CAN BE IMPAIRED BY ADVERSE FACTORS, INCLUDING BUT NOT LIMITED TO FLUCTUATIONS IN ELECTRICAL POWER SUPPLY, COMPUTER HARDWARE MALFUNCTIONS, COMPUTER OPERATING SYSTEM SOFTWARE FITNESS, FITNESS OF COMPILERS AND DEVELOPMENT SOFTWARE USED TO DEVELOP AN APPLICATION, INSTALLATION ERRORS, SOFTWARE AND HARDWARE COMPATIBILITY PROBLEMS, MALFUNCTIONS OR FAILURES OF ELECTRONIC MONITORING OR CONTROL DEVICES, TRANSIENT FAILURES OF ELECTRONIC SYSTEMS (HARDWARE AND/OR SOFTWARE), UNANTICIPATED USES OR MISUSES, OR ERRORS ON THE PART OF THE USER OR APPLICATIONS DESIGNER (ADVERSE FACTORS SUCH AS THESE ARE HEREAFTER COLLECTIVELY TERMED "SYSTEM FAILURES"). ANY APPLICATION WHERE A SYSTEM FAILURE WOULD CREATE A RISK OF HARM TO PROPERTY OR PERSONS (INCLUDING THE RISK OF BODILY INJURY AND DEATH) SHOULD NOT BE RELIANT SOLELY UPON ONE FORM OF ELECTRONIC SYSTEM DUE TO THE RISK OF SYSTEM FAILURE. TO AVOID DAMAGE, INJURY, OR DEATH, THE USER OR APPLICATION DESIGNER MUST TAKE REASONABLY PRUDENT STEPS TO PROTECT AGAINST SYSTEM FAILURES, INCLUDING BUT NOT LIMITED TO BACK-UP OR SHUT DOWN MECHANISMS. BECAUSE EACH END-USER SYSTEM IS CUSTOMIZED AND DIFFERS FROM NATIONAL INSTRUMENTS' TESTING PLATFORMS AND BECAUSE A USER OR APPLICATION DESIGNER MAY USE NATIONAL INSTRUMENTS PRODUCTS IN COMBINATION WITH OTHER PRODUCTS IN A MANNER NOT EVALUATED OR CONTEMPLATED BY NATIONAL INSTRUMENTS, THE USER OR APPLICATION DESIGNER IS ULTIMATELY RESPONSIBLE FOR VERIFYING AND VALIDATING THE SUITABILITY OF NATIONAL INSTRUMENTS PRODUCTS WHENEVER NATIONAL INSTRUMENTS PRODUCTS ARE INCORPORATED IN A SYSTEM OR APPLICATION, INCLUDING, WITHOUT LIMITATION, THE APPROPRIATE DESIGN, PROCESS AND SAFETY LEVEL OF SUCH SYSTEM OR APPLICATION.

Contents

About This Manual

Conventions xi
Related Documentation xii
NI Vision xii
NI Vision Assistant xii
NI Vision Builder for Automated Inspection xii
Other Documentation xiii

Chapter 1
Introduction to NI Vision

About NI Vision 1-1
NI Vision Control Palette 1-1
NI Vision Function Palettes 1-2
Vision Utilities 1-2
Image Processing 1-4
Machine Vision 1-5
How to Create NI Vision Applications 1-6

Chapter 2
Getting Measurement-Ready Images

Set Up Your Imaging System 2-1
Calibrate Your Imaging System 2-2
Create an Image 2-2
Input and Output Combinations 2-4
Image Analysis 2-4
Image Masks 2-4
Image Filling 2-5
Image Processing 2-5
Arithmetic and Logical Operations 2-6
Acquire or Read an Image 2-6
Display an Image 2-8
External Window Display 2-8
Image Display Control 2-9
Attach Calibration Information 2-12
Analyze an Image 2-12

Improve an Image 2-14
Lookup Tables 2-14
Filters 2-15
Convolution Filter 2-15
Nth Order Filter 2-15
Grayscale Morphology 2-15
FFT 2-16
Advanced Operations 2-18

Chapter 3
Making Grayscale and Color Measurements

Define Regions of Interest 3-1
Defining Regions Interactively 3-3
Defining an ROI in the Image Display Control 3-3
Defining an ROI in an External Window 3-4
Defining an ROI Using an ROI Constructor 3-4
Tools Palette Transformation 3-6
Defining Regions Programmatically 3-8
Defining Regions with Masks 3-9
Measure Grayscale Statistics 3-10
Measure Color Statistics 3-10
Comparing Colors 3-12
Learning Color Information 3-12
Specifying the Color Information to Learn 3-13
Choosing a Color Representation Sensitivity 3-15
Ignoring Learned Colors 3-16

Chapter 4
Performing Particle Analysis

Create a Binary Image 4-1
Improve the Binary Image 4-3
Removing Unwanted Particles 4-3
Separating Touching Particles 4-4
Using Watershed Transform 4-4
Using Binary Morphology 4-4
Improving Particle Shapes 4-5
Make Particle Measurements 4-5

Chapter 5
Performing Machine Vision Tasks

Locate Objects to Inspect 5-2
Using Edge Detection to Build a Coordinate Transformation 5-3
Using Pattern Matching to Build a Coordinate Transformation 5-6
Choosing a Method to Build the Coordinate Transformation 5-7
Set Search Areas 5-8
Defining Regions Interactively 5-8
Defining Regions Programmatically 5-9
Find Measurement Points 5-9
Finding Features Using Edge Detection 5-9
Finding Lines or Circles 5-9
Finding Edge Points Along One Search Contour 5-11
Finding Edge Points Along Multiple Search Contours 5-12
Finding Points Using Pattern Matching 5-13
Defining and Creating Effective Template Images 5-13
Training the Pattern Matching Algorithm 5-15
Defining a Search Area 5-15
Setting Matching Parameters and Tolerances 5-17
Testing the Search Algorithm on Test Images 5-18
Using a Ranking Method to Verify Results 5-18
Finding Points Using Geometric Matching 5-18
Defining and Creating Effective Template Images 5-19
Training the Geometric Matching Algorithm 5-20
Setting Matching Parameters and Tolerances 5-21
Testing the Search Algorithm on Test Images 5-22
Finding Points Using Color Pattern Matching 5-24
Defining and Creating Good Color Template Images 5-24
Training the Color Pattern Matching Algorithm 5-25
Defining a Search Area 5-26
Setting Matching Parameters and Tolerances 5-27
Testing the Search Algorithm on Test Images 5-29
Finding Points Using Color Location 5-30
Convert Pixel Coordinates to Real-World Coordinates 5-30
Make Measurements 5-31
Making Distance Measurements 5-31
Making Analytic Geometry Measurements 5-32
Making Instrument Reader Measurements 5-32
Identify Parts Under Inspection 5-33
Classifying Samples 5-33
Reading Characters 5-34

Reading Barcodes 5-35
Reading 1D Barcodes 5-35
Reading Data Matrix Barcodes 5-35
Reading PDF417 Barcodes 5-35
Inspect Image for Defects 5-36
Compare to Golden Template 5-36
Verify Characters 5-36
Display Results 5-37

Chapter 6
Calibrating Images

Perspective and Nonlinear Distortion Calibration 6-1
Defining a Calibration Template 6-2
Defining a Reference Coordinate System 6-3
Learning Calibration Information 6-5
Specifying Scaling Factors 6-6
Choosing a Region of Interest 6-6
Choosing a Learning Algorithm 6-6
Using the Learning Score 6-7
Learning the Error Map 6-8
Learning the Correction Table 6-8
Setting the Scaling Method 6-8
Calibration Invalidation 6-9
Simple Calibration 6-9
Save Calibration Information 6-10
Attach Calibration Information 6-10

Appendix A
Using NI Vision with the LabVIEW Real-Time Module

Overview A-1
System Components A-1
Development System A-1
Deployed System A-2
Software Installation A-2
Image Display A-2
Remote Display A-3
RT Video Out A-4
Determinism in Real-Time Applications A-5
Determinism versus Time-Bounded Execution A-5

Time-Bounded Execution....A-6
Initializing the Timed Environment....A-6
Preparing Resources....A-7
Performing Time-Bounded Vision Operations....A-8
Closing the Timed Environment....A-9
Image Files....A-9
Deployment....A-9
Troubleshooting....A-10
Remote Display Errors....A-10
Programming Errors....A-11
RT Video Out Errors....A-13

Appendix B
Technical Support and Professional Services

Glossary

Index

About This Manual

This document is intended for engineers and scientists who have knowledge of the LabVIEW programming environment and need to create machine vision and image processing applications using LabVIEW VIs. The manual guides you through tasks beginning with setting up your imaging system to taking measurements. It also describes how to create a real-time vision application using NI Vision with the LabVIEW Real-Time (RT) Module.

Conventions

The following conventions appear in this manual:

» The **»** symbol leads you through nested menu items and dialog box options to a final action. The sequence **File»Page Setup»Options** directs you to pull down the **File** menu, select the **Page Setup** item, and select **Options** from the last dialog box.

This icon denotes a tip, which alerts you to advisory information.

This icon denotes a note, which alerts you to important information.

bold Bold text denotes items that you must select or click in the software, such as menu items and dialog box options. Bold text also denotes parameter names.

italic Italic text denotes variables, emphasis, a cross reference, or an introduction to a key concept. Italic text also denotes text that is a placeholder for a word or value that you must supply.

`monospace` Text in this font denotes text or characters that you should enter from the keyboard, sections of code, programming examples, and syntax examples. This font is also used for the proper names of disk drives, paths, directories, programs, subprograms, subroutines, device names, functions, operations, variables, filenames, and extensions.

`monospace bold` Bold text in this font denotes the messages and responses that the computer automatically prints to the screen. This font also emphasizes lines of code that are different from the other examples.

Related Documentation

In addition to this manual, the following documentation resources are available to help you create your vision application.

NI Vision

- *NI Vision Concepts Manual*—Describes the basic concepts of image analysis, image processing, and machine vision. This document also contains in-depth discussions about imaging functions for advanced users.
- *NI Vision for LabVIEW VI Reference Help*—Contains reference information about NI Vision for LabVIEW palettes and VIs.

NI Vision Assistant

- *NI Vision Assistant Tutorial*—Describes the NI Vision Assistant software interface and guides you through creating example image processing and machine vision applications.
- *NI Vision Assistant Help*—Contains descriptions of the NI Vision Assistant features and functions and provides instructions for using them.

NI Vision Builder for Automated Inspection

- *NI Vision Builder for Automated Inspection Tutorial*—Describes Vision Builder for Automated Inspection and provides step-by-step instructions for solving common visual inspection tasks, such as inspection, gauging, part presence, guidance, and counting.
- *NI Vision Builder for Automated Inspection: Configuration Help*—Contains information about using the Vision Builder for Automated Inspection Configuration Interface to create a machine vision application.
- *NI Vision Builder for Automated Inspection: Inspection Help*—Contains information about running applications created with Vision Builder for Automated Inspection (Vision Builder AI) in the Vision Builder AI Interface.

Other Documentation

- National Instruments image acquisition device user manuals—Contain installation instructions and device-specific information.
- *Getting Started With NI-IMAQ*—Contains information about new functionality, minimum system requirements, and installation instructions for NI-IMAQ driver software.
- *NI Vision Hardware Help*—Contains fundamental programming concepts for NI-IMAQ driver software and terminology for using NI image acquisition devices.
- *NI-IMAQ VI Reference Help*—Contains reference information about the LabVIEW palettes and VIs for NI-IMAQ driver software.
- National Instruments PXI controller user manual—Contains information about how to set up your PXI controller device in your PXI chassis if you are using the LabVIEW Real-Time Module to develop your vision application.
- *LabVIEW Help*—Enter "NI-IMAQ I/O" in the **Search** tab of the *LabVIEW Help* for information about configuring the NI CVS-1450 Series compact vision system digital I/O and shutdown components as well as parameter reference information for each of the components.
- Example programs—Illustrate common applications you can create with Vision for LabVIEW. You can find these examples by launching the NI Example Finder from **Help»Find Examples** within LabVIEW 7.0 and later.
- Application Notes—Contain information about advanced NI Vision concepts and applications. Application Notes are located on the National Instruments Web site at `ni.com/appnotes.nsf`.
- NI Developer Zone (NIDZ)—Contains example programs, tutorials, technical presentations, the Instrument Driver Network, a measurement glossary, an online magazine, a product advisor, and a community area where you can share ideas, questions, and source code with developers around the world. The NI Developer Zone is located on the National Instruments Web site at `ni.com/zone`.

1

Introduction to NI Vision

This chapter describes the NI Vision for LabVIEW software, outlines the NI Vision palette organization, and lists the steps for making a machine vision application.

Note Refer to the *NI Vision Development Module Release Notes* that came with your software for information about the system requirements and installation procedures for NI Vision for LabVIEW.

About NI Vision

NI Vision for LabVIEW—a part of the NI Vision Development Module—is a library of LabVIEW VIs that you can use to develop machine vision and scientific imaging applications. The NI Vision Development Module also includes the same imaging functions for LabWindows™/CVI™ and other C development environments, as well as ActiveX controls for Visual Basic. Vision Assistant, another Vision Development Module software product, enables you to prototype your application strategy quickly without having to do any programming. Additionally, NI offers Vision Builder AI: configurable machine vision software that you can use to prototype, benchmark, and deploy applications.

NI Vision Control Palette

The Vision control palette is available from the top level of the controls palette in LabVIEW 7.0 or later. The Vision control palette has the following elements:

- IMAQ Image.ctl—This control is the type definition that describes the image data type. You can use this control to represent the image data type on the front panel of a VI. For example, use this control as an input or output of a subVI so that a calling VI can pass an image to the subVI.

- Image Display—Use this control to display your images directly on the front panel of LabVIEW 7.0 or later. You also can use this control to create regions of interest (ROIs). Classic and 3D versions are available.

- NI Vision controls—Use these controls to get the functionality of corresponding NI Vision VI controls directly into your own VIs.

- Machine Vision controls—Use these controls to get the functionality of corresponding Machine Vision VI controls directly into your own VIs.

NI Vision Function Palettes

NI Vision for LabVIEW is organized into three main function palettes: Vision Utilities, Image Processing, and Machine Vision. This section describes these palettes and their subpalettes.

Note This document references many VIs from the NI Vision function palette. If you have difficulty finding a VI, use the search capability of the LabVIEW VI browser.

Vision Utilities

Vision Utilities functions allow you to manipulate and display images in NI Vision.

- Image Management—A group of VIs that manage images. Use these VIs to create and dispose images, set and read attributes of an image such as its size and offset, and copy one image to another. You also can use some of the advanced VIs to define the border region of an image and access the pointer to the image data.

- Files—A group of VIs that read images from files, write images to files in different file formats, and get information about the image contained in a file.

- External Display—A group of VIs that control the display of images in external image windows. Use these VIs to complete the following tasks:
  - Get and set window attributes, such as size, position, and zoom factor
  - Assign color palettes to image windows
  - Set up and use image browsers
  - Set up and use different drawing tools to interactively select ROIs on image windows
  - Detect draw events
  - Retrieve information about ROIs drawn on the image window

Note If you have LabVIEW 7.0 or later, you also can use the Image Display control available from the Vision control palette.

- Region of Interest—A group of VIs that manage ROIs. Use these VIs to programmatically define ROIs and convert ROIs to and from image masks.

Note If you have LabVIEW 7.0 or later, you can use the property node and invoke node of the Image Display control to perform many of these ROI tasks.

- Image Manipulation—A group of VIs that modify the spatial content of images. Use these VIs to resample an image, extract parts of an image, and rotate, shift, and unwrap images. This subpalette also contains VIs that copy images to and from the clipboard.

- Pixel Manipulation—A group of VIs that read and modify individual pixels in an image. Use these VIs to read and set pixel values in an image or along a row or column in an image, fill the pixels in an image with a particular value, and convert an image to and from a 2D LabVIEW array.

- Overlay—A group of VIs that overlay graphics on an *image display environment* without altering the pixel values of the image. Use these VIs to overlay the results of your inspection application onto the images you inspect.

- Calibration—A group of VIs that spatially calibrate an image to take accurate, real-world measurements regardless of camera perspective or lens distortion. Use these VIs to set a simple calibration or to let NI Vision automatically learn the calibration data from a grid image. Then use the VIs to convert pixel coordinates to real-world coordinates for simple measurements.

- Color Utilities—A group of VIs that access data from color images. Use these VIs to extract different color planes from an image, replace the planes of a color image with new data, convert a color image to and from a 2D array, read and set pixel values in a color image, and convert pixel values from one color space to another.

- Vision RT—A group of VIs that provide functionality for using NI-IMAQ and NI Vision with the LabVIEW Real-Time (RT) Module. Use these VIs to display images to Video Out on your RT system, to control the compression setting for sending images over the network, and to time bound your processing VIs on your RT system.

Image Processing

Use the Image Processing functions to analyze, filter, and process images in NI Vision.

- Processing—A group of VIs that process grayscale and binary images. Use these VIs to convert a grayscale image into a binary image using different thresholding techniques. You also can use these VIs to transform images using predefined or custom lookup tables, apply a watershed transform, change the contrast information in the image, and invert the values in an image.

- Filters—A group of VIs that filter an image to enhance the information in the image. Use these VIs to smooth an image, remove noise, and highlight or enhance edges in the image. You can use a predefined convolution kernel or create custom convolution kernels.

- Morphology—A group of VIs that perform morphological operations on an image. Some of these VIs perform basic morphological operations, such as dilation and erosion, on grayscale and binary images. Other VIs improve the quality of binary images by filling holes in particles, removing particles that touch the image border, removing small particles, and removing unwanted particles based on different shape characteristics of the particle. Another set of VIs in this subpalette separate touching particles, find the skeleton of particles, and detect circular particles.

- Analysis—A group of VIs that analyze the content of grayscale and binary images. Use these VIs to compute the *histogram* information and grayscale statistics of an image, retrieve pixel information and statistics along any one-dimensional profile in an image, and detect and measure particles in binary images.

- Color Processing—A group of VIs that analyze and process color images. Use these VIs to compute the histogram of color images; apply *lookup tables* to color images; change the brightness, contrast, and gamma information associated with a color image; and threshold a color image. Some of these VIs also compare the color information in different images or different regions in an image using a color matching process.

- Operators—A group of VIs that perform basic arithmetic and logical operations on images. Use some of these VIs to add, subtract, multiply, and divide an image with other images or constants. Use other VIs in this subpalette to apply logical operations—such as AND/NAND, OR/NOR, XOR/XNOR—and make pixel comparisons between an image and other images or a constant. In addition, one VI in this

subpalette allows you to select regions in an image to process using a masking operation.

- Frequency Domain—A group of VIs that analyze and process images in the frequency domain. Use these VIs to convert an image from the spatial domain to the frequency domain using a two-dimensional Fast Fourier Transform (FFT) and convert from the frequency domain to the spatial domain using the inverse FFT. These VIs also extract the magnitude, phase, real, and imaginary planes of the complex image. In addition, these VIs allow you to convert complex images into complex 2D arrays and back. Also in this subpalette are VIs that perform basic arithmetic operations—such as addition, subtraction, multiplication, and division—between a complex image and other images or a constant. Lastly, some of these VIs allow you to filter images in the frequency domain.

Machine Vision

The NI Machine Vision VIs are high-level VIs that simplify common machine vision tasks.

- Select Region of Interest—A group of VIs that allow you to select a ROI tool, draw specific ROIs in the image window, and return information about ROIs with very little programming.

- Coordinate System—A group of VIs that find a coordinate system associated with an object in an image. Use these VIs to find the coordinate system using either edge detection or pattern matching. You can then use this coordinate system to take measurements from other Machine Vision VIs.

- Count and Measure Objects—A VI that thresholds an image to isolate objects from the background and then finds and measures characteristics of the objects. This VI also can ignore unwanted objects in the image when making measurements.

- Measure Intensities—A group of VIs that measure the intensity of a pixel at a point or the statistics of pixel intensities along a line or rectangular region in an image.

- Measure Distances—A group of VIs that measure distances, such as the minimum and maximum horizontal distance between two vertically oriented edges or the minimum and maximum vertical distance between two horizontally oriented edges.

- Locate Edges—A group of VIs that locate vertical, horizontal, and circular edges.

- Find Patterns—A VI that learns and searches for a pattern in an image.

- Searching and Matching—A group of VIs that create and search for patterns in grayscale and color images. This subpalette also contains a VI to search for objects with predefined shapes in binary images.

- Caliper—A group of VIs that find edges along different profiles in the image. Use these VIs to find edges along a line, a set of parallel lines defined inside a rectangular region (rake), a set of parallel concentric lines defined inside an annular region (concentric rake), or a set of radial lines defined inside an annular region (spoke). You also can use these VIs to find pairs of edges in the image that satisfy certain criteria.

- Analytic Geometry—A group of VIs that perform analytic geometry computations on a set of points in an image. Use these VIs to fit lines, circles, and ellipses to a set of points in the image; compute the area of a polygon represented by a set of points; measure distances between points; and find angles between lines represented by points. VIs in this subpalette also perform computations, such as finding the intersection point of two lines and finding the line bisecting the angle formed by two lines.

- OCR—A group of VIs that perform optical character recognition in a region of the image.

- Classification—A group of VIs that classify binary objects according to their shape or any user-defined feature vector.

- Instrument Readers—A group of VIs that accelerate the development of applications that require reading from seven-segment displays, meters, gauges, 1D barcodes, or 2D barcodes.

- Inspection—A group of VIs that compare images to a golden template image.

How to Create NI Vision Applications

Figures 1-1 and 1-2 illustrate the steps for creating an application with NI Vision. Figure 1-1 describes the general steps to designing an NI Vision application. Items 2 and 3 in the last step in Figure 1-1 are expanded upon in Figure 1-2. You can use a combination of the items in the last step to create your NI Vision application. Refer to the corresponding chapter beside a figure item for more information about the item.

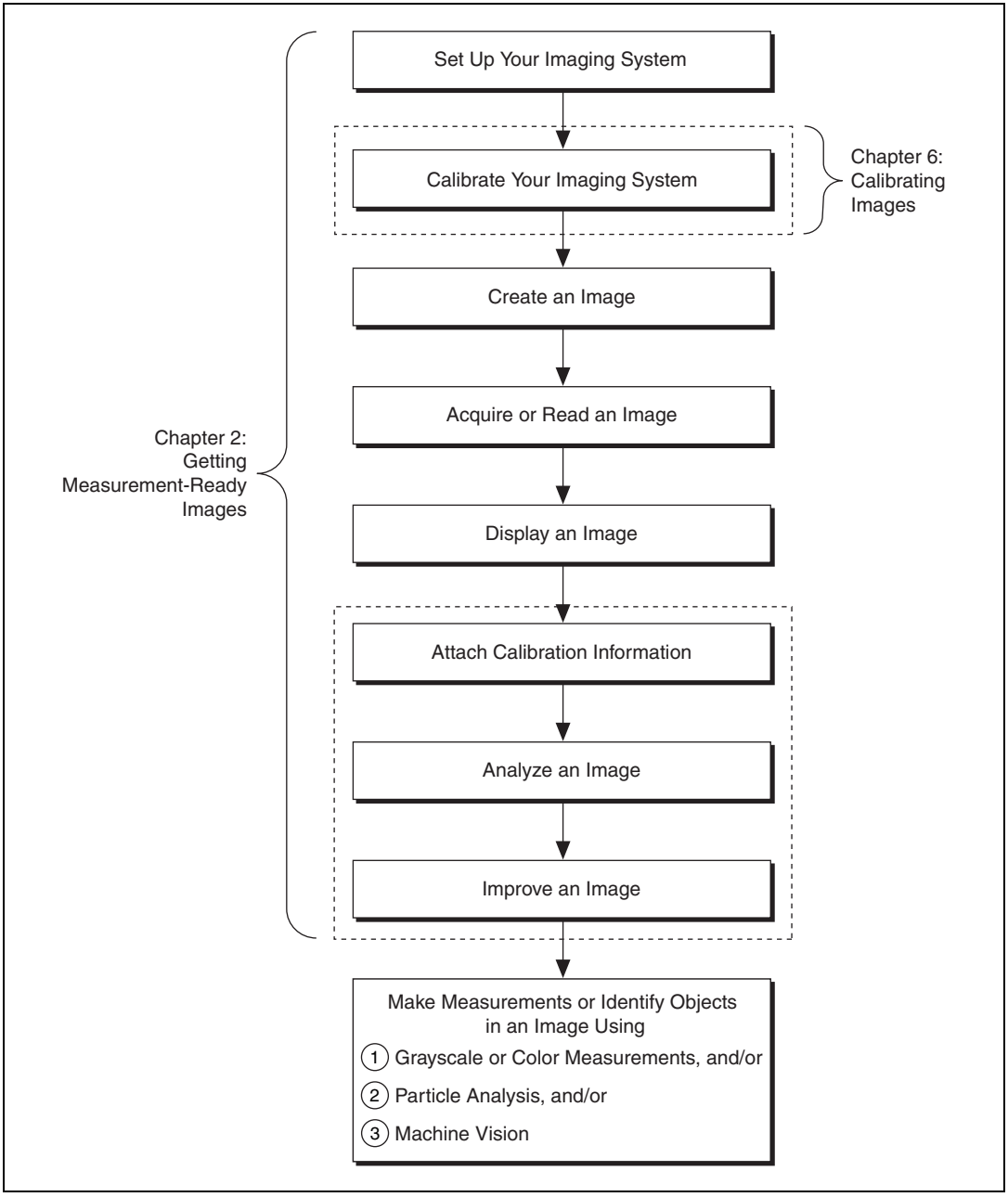

Set Up Your Imaging System
Calibrate Your Imaging System
Chapter 6: Calibrating Images
Create an Image
Acquire or Read an Image
Chapter 2: Getting Measurement-Ready Images
Display an Image
Attach Calibration Information
Analyze an Image
Improve an Image
Make Measurements or Identify Objects in an Image Using
1 Grayscale or Color Measurements, and/or
2 Particle Analysis, and/or
3 Machine Vision

Figure 1-1. General Steps for Designing a Vision Application

Note Diagram items enclosed with dashed lines are optional steps.

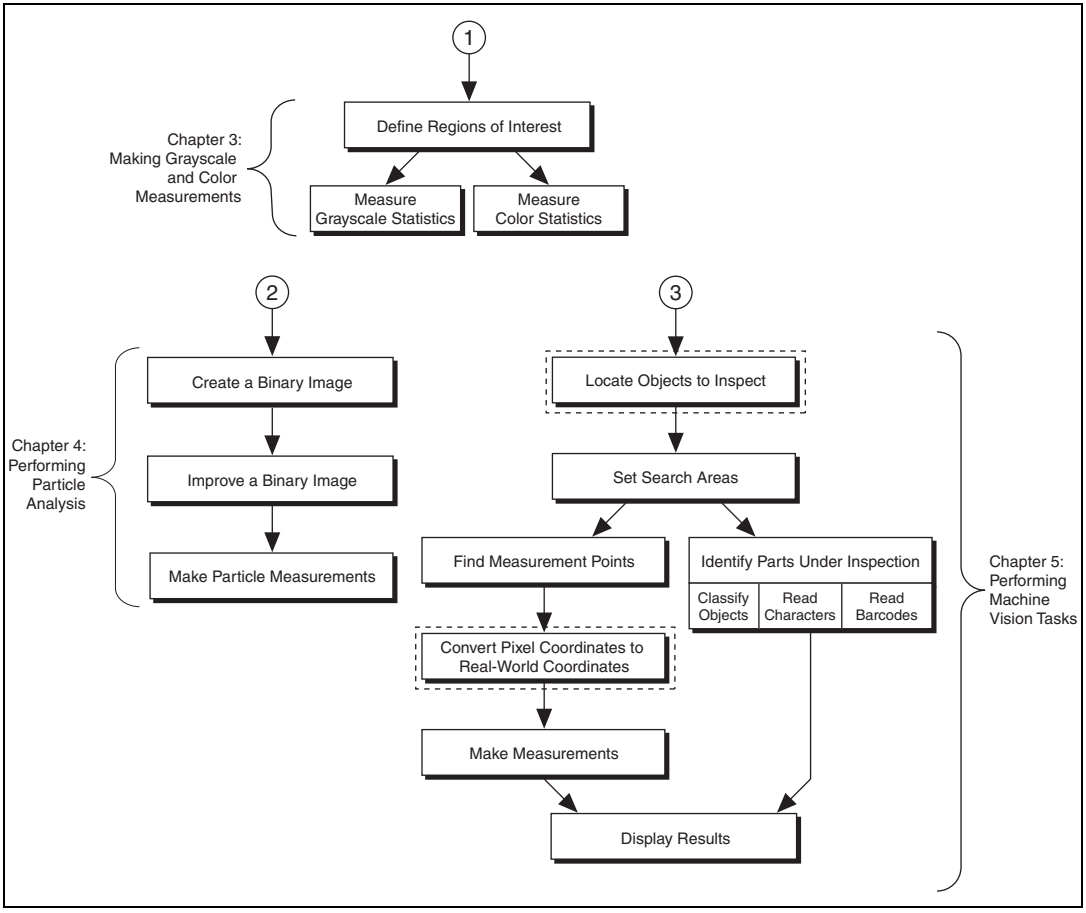


Figure 1-2. Inspection Steps for Building a Vision Application

Note Diagram items enclosed with dashed lines are optional steps.

2

Getting Measurement-Ready Images

This chapter describes how to set up your imaging system, acquire and display an image, analyze the image, and prepare the image for additional processing.

Set Up Your Imaging System

Before you acquire, analyze, and process images, you must set up your imaging system. The manner in which you set up your system depends on your imaging environment and the type of analysis and processing you need to do. Your imaging system should produce images with high enough quality so that you can extract the information you need from the images.

Complete the following steps to set up your imaging system.

1. Determine the type of equipment you need given space constraints and the size of the object you need to inspect. Refer to Chapter 3, *System Setup and Calibration*, of the *NI Vision Concepts Manual* for more information.
    a. Make sure your camera sensor is large enough to satisfy your minimum resolution requirement.
    b. Make sure your lens has a depth of field high enough to keep all of your objects in focus regardless of their distance from the lens. Also, make sure your lens has a focal length that meets your needs.
    c. Make sure your lighting provides enough contrast between the object under inspection and the background for you to extract the information you need from the image.
2. Position your camera so that it is perpendicular to the object under inspection. If your camera acquires images of the object from an angle, perspective errors occur. Even though you can compensate for these errors with software, NI recommends that you use a perpendicular inspection angle to obtain the most accurate results.

3. Select an image acquisition device that meets your needs. National Instruments offers several image acquisition devices, such as analog color and monochrome devices as well as digital devices. Visit ni.com/vision for more information about NI image acquisition devices.
4. Configure the driver software for your image acquisition device. If you have an NI image acquisition device, configure your NI-IMAQ driver software through MAX. Open MAX by double-clicking the Measurement & Automation Explorer icon on your desktop. Refer to the *NI Vision Hardware Help* and the *NI-IMAQ Help* for more information.

Calibrate Your Imaging System

After you set up your imaging system, you may want to calibrate your system to assign real-world coordinates to pixel coordinates and compensate for perspective and nonlinear errors inherent in your imaging system.

Perspective errors occur when your camera axis is not perpendicular to the object under inspection. Nonlinear distortion may occur from aberrations in the camera lens. Perspective errors and lens aberrations cause images to appear distorted. This distortion displaces information in an image, but it does not necessarily destroy the information in the image.

Use simple calibration if you only want to assign real-world coordinates to pixel coordinates. Use perspective and nonlinear distortion calibration if you need to compensate for perspective errors and nonlinear lens distortion. Refer to Chapter 6, *Calibrating Images*, for detailed information about calibration.

Create an Image

Use the IMAQ Create VI (**Vision Utilities»Image Management**) to create an image reference. When you create an image, specify one of the following image data types:

- Grayscale (U8, default)—8-bit unsigned
- Grayscale (I16)—16-bit signed
- Grayscale (SGL)—Floating point
- Complex (CSG)

- RGB (U32)—32-bit RGB
- HSL (U32)—32-bit HSL
- RGB (U64)—64-bit RGB

You can create multiple images by executing IMAQ Create as many times as you want, but each image you create requires a unique name. Determine the number of required images through an analysis of your intended application. Base your decision on different processing phases and whether you need to keep the original image after each processing phase.

Note If you plan to use filtering or particle analysis VIs on the image, refer to their help topics in the *NI Vision for LabVIEW VI Reference Help* for information about the appropriate border size for the image. The default border size is three pixels.

When you create an image, NI Vision creates an internal image structure to hold properties of the image, such as its name and border size. However, no memory is allocated to store the image pixels at this time. NI Vision VIs automatically allocate the appropriate amount of memory when the image size is modified. For example, VIs that acquire or resample an image alter the image size so they allocate the appropriate memory space for the image pixels. The output of IMAQ Create is a reference to the image structure. Supply this reference as an input to all subsequent NI Vision functions.

During development, you may want to examine the contents of your image at run time. With LabVIEW 7.0 or later, you can use a LabVIEW image probe to view the contents of your image during execution. To create a probe, right-click on the image wire and select **Probe**.

Most VIs belonging to the NI Vision library require an input of one or more image references. The number of image references a VI takes depends on the image processing function and the type of image you want to use.

NI Vision VIs that analyze the image but do not modify the contents require the input of only one image reference. VIs that process the contents of images may require a reference to the source image(s) and to a destination image, or the VIs may have an optional destination image. If you do not provide a destination image, the VI modifies the source image.

At the end of your application, dispose of each image that you created using the IMAQ Dispose VI (**Vision Utilities»Image Management**).

Input and Output Combinations

Depending on the type of function a VI performs, different combinations of input and output are possible. You can use this flexibility to decide which image to process and where to store the resulting image. If no destination image is wired, the source image is used and passed to the destination output.

The figures in the following sections show several VI connector panes used in NI Vision.

Image Analysis

The following connector pane applies only to VIs that analyze an image and therefore do not modify either the size or contents of the image. Examples of these types of operations include particle analysis and *histogram* calculations.

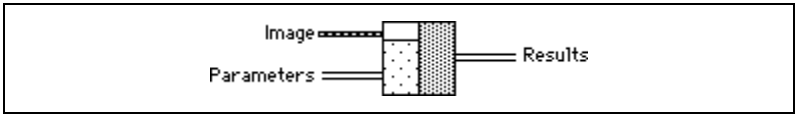
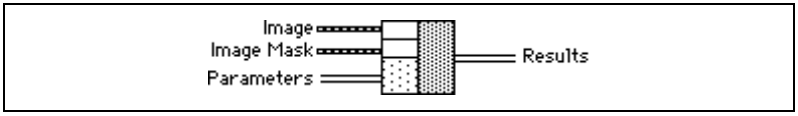


Image Masks

The following connector pane introduces an **Image Mask**.

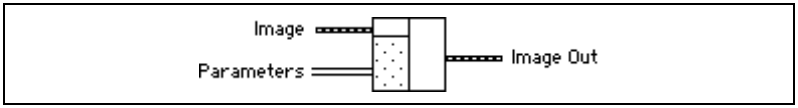


The presence of an **Image Mask** input indicates that the processing or analysis is dependent on the contents of another image: the **Image Mask**. The only pixels in **Image** that are processed are those whose corresponding pixels in **Image Mask** are non-zero. If an **Image Mask** pixel is 0, the corresponding **Image** pixel is not changed. **Image Mask** must be an 8-bit image.

If you want to apply a processing or analysis function to the entire image, do not connect the **Image Mask** input. Connecting the same image to both inputs **Image** and **Image Mask** also gives the same effect as leaving the input **Image Mask** unconnected, except in this case the **Image** must be an 8-bit image.

Image Filling

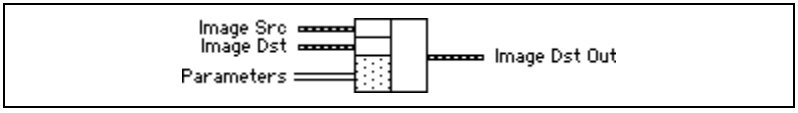
The following connector pane applies to VIs performing an operation that fills an image.



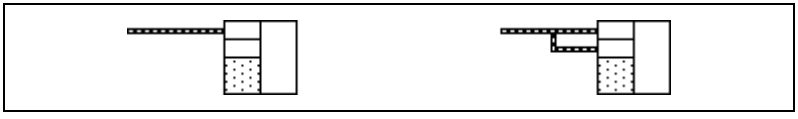
Examples of this type of operation include reading a file, acquiring an image from an NI image acquisition device, or transforming a 2D array into an image. This type of VI can modify the size of an image.

Image Processing

The following connector pane applies to VIs that process an image.



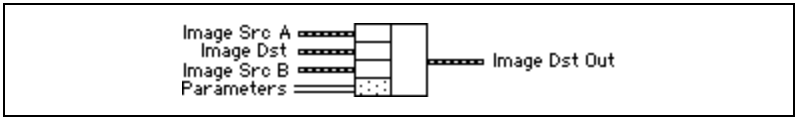
This connector is the most common type in NI Vision. The **Image Src** input receives the image to process. The **Image Dst** input can receive either another image or the original, depending on your goals. If two different images are connected to the two inputs, the original **Image Src** image is not modified. As shown in the following diagrams, if the **Image Dst** and **Image Src** inputs receive the same image, or if nothing is connected to **Image Dst**, the processed image is placed into the original image, and the original image data is lost.



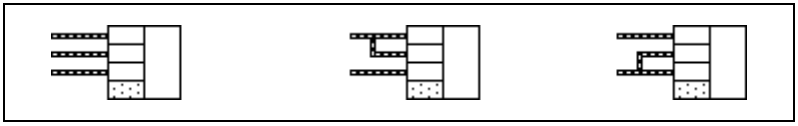
The **Image Dst** input is the image that receives the processing results. Depending on the functionality of the VI, this image can be either the same or a different image type as that of the source image. The VI descriptions in the *NI Vision for LabVIEW VI Reference Help* include the type of image that can be connected to the **Image** inputs. The image connected to **Image Dst** is resized to the source image size.

Arithmetic and Logical Operations

The following connector pane applies to VIs that perform arithmetic or logical operations between two images.



Two source images exist for the destination image. You can perform an operation between two images, A and B, and then either store the result in another image, **Image Dst**, or in one of the two source images, A or B. In the latter case, you can consider the original data to be unnecessary after the processing has occurred. The following combinations are possible in this pane.



In the pane on the left, the three images are all different. **Image Src A** and **Image Src B** are intact after processing and the results from this operation are stored in **Image Dst**.

In the center pane, **Image Src A** also is connected to the **Image Dst**, which therefore receives the results from the operation. In this operation, the source data for **Image Src A** is overwritten.

In the pane on the right, **Image Src B** receives the results from the operation and its source data is overwritten.

Most operations between two images require that the images have the same type and size. However, arithmetic operations can work between two different types of images.

Acquire or Read an Image

After you create an image reference, you can acquire an image into your imaging system in three ways. You can acquire an image with a camera through your image acquisition device, load an image from a file stored on your computer, or convert the data stored in a 2D array to an image. VIs that acquire images, load images from file, or convert data from a 2D array

automatically allocate the memory space required to accommodate the image data.

Use one of the following methods to acquire images with a National Instruments image acquisition device:

- Acquire a single image using the IMAQ Snap VI (**Image Acquisition**). When you call this VI, it initializes the image acquisition device and acquires the next incoming video frame. Use this VI for single capture applications where ease of programming is essential.

- Acquire images continually through a grab acquisition. Grab functions perform an acquisition that loops continually on one buffer. Use the grab functions for high-speed image acquisition. Use the IMAQ Grab Setup VI (**Image Acquisition**) to start the acquisition. Use the IMAQ Grab Acquire VI (**Image Acquisition**) to return a copy of the current image. Use the IMAQ Stop VI (**Image Acquisition»Low-Level Acquisition**) to stop the acquisition.

- Acquire a fixed number of images using the IMAQ Sequence VI (**Image Acquisition**). IMAQ Sequence acquires one image after another until it has acquired the number of images you requested. If you want to acquire only certain images, supply IMAQ Sequence with a table describing the number of frames to skip after each acquired frame.

Note When you are finished with the image acquisition, you must use the IMAQ Close VI (**Image Acquisition**) to release resources associated with the image acquisition device.

Use the IMAQ ReadFile VI (**Vision Utilities»Files**) to open and read data from a file stored on your computer into the image reference. You can read from image files stored in a standard format—such as BMP, TIFF, JPEG, JPEG2000, PNG, and AIPD—or a nonstandard format you specify. In all cases, the software automatically converts the pixels it reads into the type of image you pass in.

Use the IMAQ Read Image and Vision Info VI (**Vision Utilities»Files**) to open an image file containing additional information, such as calibration information, template information for pattern matching, or overlay information. Refer to Chapter 5, *Performing Machine Vision Tasks*, for information about pattern matching templates and overlays.

You also can use the IMAQ GetFileInfo VI (**Vision Utilities»Files**) to retrieve image properties—image size, pixel depth, recommended image type, and calibration units—without actually reading all the image data.

Use IMAQ AVI Open and IMAQ AVI Read Frame to open and read data from an AVI file stored on your computer into the image reference. NI Vision automatically converts the pixels it reads into the type of image you pass in.

Note When you are finished with the AVI file, you must use IMAQ AVI Close to release resources associated with the AVI file.

Use the IMAQ ArrayToImage VI (**Vision Utilities»Pixel Manipulation**) to convert a 2D array to an image. You also can use the IMAQ ImageToArray VI (**Vision Utilities»Pixel Manipulation**) to convert an image to a LabVIEW 2D array.

Display an Image

You can display images in LabVIEW using two methods. You can display an image in an external window using the external display VIs on the External Display function palette. You can also display an image directly on the front panel using the Image Display control on the Vision control palette.

External Window Display

Display an image in an external window using the IMAQ WindDraw VI (**Vision Utilities»Display**). You can display images in 16 different external windows. Use the IMAQ WindSetup VI (**Vision Utilities»Display**) to configure the appearance of each external window. For example, you can decide if the window has scroll bars, is resizable, or has a title bar. You also can use the IMAQ WindMove VI (**Vision Utilities»Display**) to position the external image window at a particular location on the monitor.

Note External image windows are not LabVIEW panels. They are managed directly by NI Vision.

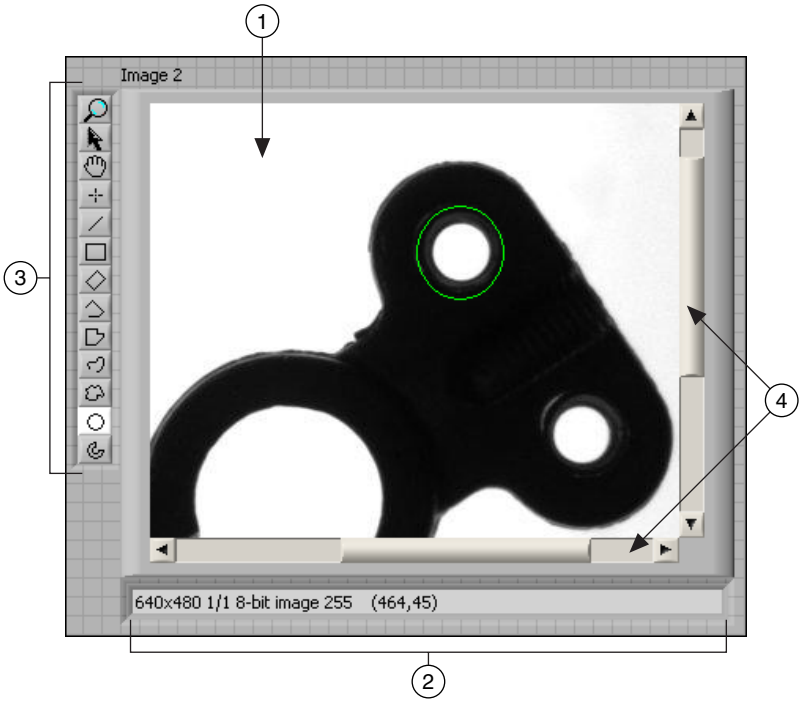
You can use a color palette to display grayscale images by applying a color palette to the window. You can use the IMAQ GetPalette VI (**Vision Utilities»Display**) to obtain predefined color palettes. For example, if you need to display a binary image—an image containing particle regions with pixel values of 1 and a background region with pixel values of 0—apply the predefined binary palette. Refer to Chapter 2, *Display*, of the *NI Vision Concepts Manual* for more information about color palettes.

Note At the end of your application, you must close all open external windows using the IMAQ WindClose VI (**Vision Utilities»Display**).

Image Display Control

Use the Image Display control to display an image on the LabVIEW front panel, as shown in Figure 2-1. You can access the Image Display control by right-clicking the front panel and selecting **Vision**.

Note If your Palette View is set to **Express**, you can access the Image Display control by right-clicking the front panel and selecting **All Controls»Vision**.



1	Display Area	3	ROI Tools Palette
2	Image Information Indicator	4	Scrollbars

Figure 2-1. LabVIEW Image Display Control

To display an image, wire the image output of an NI Vision VI into the image display terminal on the block diagram, as shown in Figure 2-2.

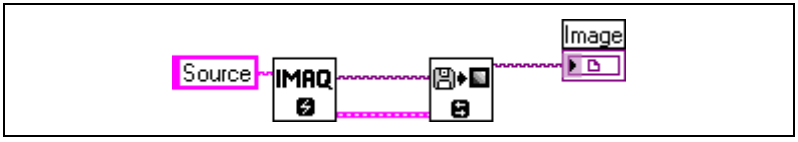


Figure 2-2. An Image Wired into the Image Display Control Terminal

The Image Display control contains the following elements:

- Display area—Displays an image.
- Image information indicator—Displays information about your image and the ROI that you are currently drawing.

- ROI tools palette—Contains tools for drawing ROIs, panning, and zooming. Unlike external display windows, each Image Display control uses its own set of tools.
- Scrollbars—Allows you to position the image in the display area.

During design time, you can customize the appearance of the control by rearranging the control elements, configuring properties through the popup menu, or selecting the control and clicking **Edit»Customize Control**.

During run time, you can customize many parts of the control using property nodes.

Note Not all functionality available during design time is available at run-time.

To create a property node, right-click the control and select **Create» Property Node**. Click the Property Node once to see the properties you can configure. Properties specific to the Image Display control appear at the end of the list.

The following list describes a subset of the properties available for the Image Display control:

- Snapshot Mode—Determines if the control makes a copy of the image or has a reference to the image. When you enable the Snapshot Mode, if the inspection image changes later in your application, the Image Display control continues to display the image as it was when the image was wired into the Image Display control.

  Enabling the Snapshot Mode may reduce the speed of your application because the control makes a copy of the image. Enable this property when you want to display a snapshot of the image in time. Disable this property when you need to display results quickly, such as during a grab acquisition. The property is disabled by default.

Note To cause the Image Display control to refresh the image immediately, use the **Refresh Image** method. To create a method, right-click the control, and select **Create» Invoke Node**. Click the Invoke Node once to see the available methods. Methods specific to the Image Display control appear at the end of the popup menu.

- Palette—Determines which color palette the Image Display control uses to display images. You can configure the control to use a predefined color palette or a custom color palette. Define a custom color palette with the **User Palette** property node. You also can change the color palette of the control or an image probe at run-time by right-clicking the Image Display control.

- Maximum Contour Count—Sets the maximum number of ROI contours a user can draw on an image display.

The Image Display control also includes the following methods:

- Get Last Event—Returns the last user event, resulting from mouse movements and clicks, on the Image Display control. This method has the same behavior as IMAQ WindLastEvent does for external display windows.
- Clear ROI—Removes any ROIs on the Image Display control.
- Refresh Image—Refreshes the display to show the latest image. This method is useful if the snapshot control is disabled, but you want the Image Display control to show the latest changes to the image.

Attach Calibration Information

If you want to attach the calibration information of the current setup to each image you acquire, use the IMAQ Set Calibration Info VI (**Vision Utilities»Calibration**). This VI takes in a source image containing the calibration information and a destination image that you want to calibrate. The output image is your inspection image with the calibration information attached to it. Refer to Chapter 6, *Calibrating Images*, for detailed information about calibration.

Note Because calibration information is part of the image, it is propagated throughout the processing and analysis of the image. Functions that modify the image size—such as geometrical transforms—void the calibration information. Use IMAQ Write Image and Vision Info (**Vision Utilities»Calibration**) to save the image and all of the attached calibration information to a file.

Analyze an Image

When you acquire and display an image, you may want to analyze the contents of the image for the following reasons:

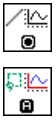
- To determine if the image quality is sufficient for your inspection task
- To obtain the values of parameters that you want to use in processing functions during the inspection process

The histogram and line profile tools can help you analyze the quality of your images.

Use the IMAQ Histograph and IMAQ Histogram VIs (**Image Processing»Analysis**) to analyze the overall grayscale distribution in the image. Use the histogram of the image to analyze two important criteria that define the quality of an image: saturation and contrast. If your image is underexposed because it was acquired in an environment without sufficient light, the majority of your pixels will have low intensity values, which appear as a concentration of peaks on the left side of your histogram. If your image is overexposed because it was acquired in an environment with too much light, the majority of the pixels will have high intensity values, which appear as a concentration of peaks on the right side of your histogram. If your image has an appropriate amount of contrast, your histogram will have distinct regions of pixel concentrations. Use the histogram information to decide if the image quality is sufficient enough to separate objects of interest from the background.

If the image quality meets your needs, use the histogram to determine the range of pixel values that correspond to objects in the image. You can use this range in processing functions, such as determining a threshold range during particle analysis.

If the image quality does not meet your needs, try to improve the imaging conditions to get the necessary image quality. You may need to re-evaluate and modify each component of your imaging setup: lighting equipment and setup, lens tuning, camera operation mode, and acquisition device parameters. If you reach the best possible conditions with your setup but the image quality still does not meet your needs, try to improve the image quality using the image processing techniques described in the *Improve an Image* section of this chapter.



Use the IMAQ LineProfile VI (**Image Processing»Analysis**) to get the pixel distribution along a line in the image, or use the IMAQ ROIProfile VI (**Image Processing»Analysis**) to get the pixel distribution along a one-dimensional path in the image. To use a line profile to analyze your image, draw or specify a line across the boundary of an object in your image. Use IMAQ LineProfile to examine the pixel values along the line. By looking at the pixel distribution along this line, you can determine if the image quality is good enough to provide you with sharp edges at object boundaries. Also, you can determine if the image is noisy, and identify the characteristics of the noise.

If the image quality meets your needs, use the pixel distribution information to determine some parameters of the inspection functions you want to use. For example, use the information from the line profile to determine the strength of the edge at the boundary of an object. You can

input this information into the IMAQ Edge Tool VI (**Machine Vision» Caliper**) to find the edges of objects along the line.

Improve an Image

Using the information you gathered from analyzing your image, you may want to improve the quality of your image for inspection. You can improve your image with lookup tables, filters, grayscale morphology, and Fast Fourier transforms.

Lookup Tables

Apply lookup table (LUT) transformations to highlight image details in areas containing significant information at the expense of other areas. A LUT transformation converts input grayscale values in the source image into other grayscale values in the transformed image. NI Vision provides four VIs that directly or indirectly apply lookup tables to images:

- IMAQ MathLookup (**Image Processing»Processing**)—Converts the pixel values of an image by replacing them with values from a predefined lookup table. NI Vision has seven predefined lookup tables based on mathematical transformations. Refer to Chapter 5, *Image Processing*, of the *NI Vision Concepts Manual* for more information about these lookup tables.

- IMAQ UserLookup (**Image Processing»Processing**)—Converts the pixel values of an image by replacing them with values from a user-defined lookup table.

- IMAQ Equalize (**Image Processing»Processing**)—Distributes the grayscale values evenly within a given grayscale range. Use IMAQ Equalize to increase the contrast in images containing few grayscale values.

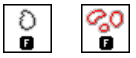
- IMAQ Inverse (**Image Processing»Processing**)—Inverts the pixel intensities of an image to compute the negative of the image. For example, use IMAQ Inverse before applying an automatic threshold to your image if the background pixels are brighter than the object pixels.

Filters

Filter your image when you need to improve the sharpness of transitions in the image or increase the overall signal-to-noise ratio of the image. You can choose either a lowpass or highpass filter depending on your needs.

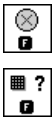
Lowpass filters remove insignificant details by smoothing the image, removing sharp details, and smoothing the edges between the objects and the background. You can use the IMAQ LowPass VI (**Image Processing»Filters**) or define your own lowpass filter with the IMAQ Convolute VI or IMAQ NthOrder VI.

Highpass filters emphasize details, such as edges, object boundaries, or cracks. These details represent sharp transitions in intensity value. You can define your own highpass filter with IMAQ Convolute or IMAQ NthOrder or use the IMAQ EdgeDetection VI or IMAQ CannyEdgeDetection VI (**Image Processing»Filters**). IMAQ EdgeDetection allows you to find edges in an image using predefined edge detection kernels, such as the Sobel, Prewitt, and Roberts kernels.



Convolution Filter

IMAQ Convolute (**Image Processing»Filters**) allows you to use a predefined set of lowpass and highpass filters. Each filter is defined by a kernel of coefficients. Use the IMAQ GetKernel VI (**Image Processing» Filters**) to retrieve predefined kernels. If the predefined kernels do not meet your needs, define your own custom filter using a LabVIEW 2D array of floating point numbers.



Nth Order Filter

IMAQ NthOrder (**Image Processing»Filters**) allows you to define a lowpass or highpass filter depending on the value of *N* that you choose. One specific Nth order filter, the median filter, removes speckle noise, which appears as small black and white dots. Refer to Chapter 5, *Image Processing*, of the *NI Vision Concepts Manual* for more information about Nth order filters.

Grayscale Morphology

Perform grayscale morphology when you want to filter grayscale features of an image. Grayscale morphology helps you remove or enhance isolated features, such as bright pixels on a dark background. Use these transformations on a grayscale image to enhance non-distinct features before thresholding the image in preparation for particle analysis.

Grayscale morphological transformations compare a pixel to those pixels surrounding it. The transformation keeps the smallest pixel values when performing an erosion or keeps the largest pixel values when performing a dilation.

Refer to Chapter 5, *Image Processing*, of the *NI Vision Concepts Manual* for more information about grayscale morphology transformations.

Use the IMAQ GrayMorphology VI (**Image Processing»Morphology**) to perform one of the following seven transformations:

- Erosion—Reduces the brightness of pixels that are surrounded by neighbors with a lower intensity. Define the neighborhood with a structuring element. Refer to Chapter 9, *Binary Morphology*, of the *NI Vision Concepts Manual* for more information about structuring elements.
- Dilation—Increases the brightness of pixels surrounded by neighbors with a higher intensity. A dilation has the opposite effect of an erosion.
- Opening—Removes bright pixels isolated in dark regions and smooths boundaries.
- Closing—Removes dark spots isolated in bright regions and smooths boundaries.
- Proper-opening—Removes bright pixels isolated in dark regions and smooths the boundaries of regions.
- Proper-closing—Removes dark pixels isolated in bright regions and smooths the boundaries of regions.
- Auto-median—Generates simpler particles that have fewer details.

FFT

Use FFT to convert an image into its frequency domain. In an image, details and sharp edges are associated with mid to high spatial frequencies because they introduce significant gray-level variations over short distances. Gradually varying patterns are associated with low spatial frequencies.

An image can have extraneous noise, such as periodic stripes, introduced during the digitization process. In the frequency domain, the periodic pattern is reduced to a limited set of high spatial frequencies. Also, the imaging setup may produce non-uniform lighting of the field of view, which produces an image with a light drift superimposed on the information you want to analyze. In the frequency domain, the light drift appears as a limited set of low frequencies around the average intensity of the image (DC component).

You can use algorithms working in the frequency domain to isolate and remove these unwanted frequencies from your image. Complete the following steps to obtain an image in which the unwanted pattern has disappeared but the overall features remain:

1. Use the IMAQ FFT VI (**Image Processing»Frequency Domain**) to convert an image from the spatial domain to the frequency domain. This VI computes the FFT of the image and results in a complex image representing the frequency information of your image.
2. Improve your image in the frequency domain with a lowpass or highpass frequency filter. Specify which type of filter to use with the IMAQ ComplexAttenuate VI (**Image Processing»Frequency Domain**) or the IMAQ ComplexTruncate VI (**Image Processing» Frequency Domain**). Lowpass filters smooth noise, details, textures, and sharp edges in an image. Highpass filters emphasize details, textures, and sharp edges in images, but they also emphasize noise.

   - Lowpass attenuation—The amount of attenuation is directly proportional to the frequency information. At low frequencies, there is little attenuation. As the frequencies increase, the attenuation increases. This operation preserves all of the zero frequency information. Zero frequency information corresponds to the DC component of the image or the average intensity of the image in the spatial domain.
   - Highpass attenuation—The amount of attenuation is inversely proportional to the frequency information. At high frequencies, there is little attenuation. As the frequencies decrease, the attenuation increases. The zero frequency component is removed entirely.
   - Lowpass truncation—Frequency components above the ideal cutoff frequency are removed, and the frequencies below it remain unaltered.
   - Highpass truncation—Frequency components above the ideal cutoff frequency remain unaltered, and the frequencies below it are removed.
3. To transform your image back to the spatial domain, use the IMAQ InverseFFT VI (**Image Processing»Frequency Domain**).

Advanced Operations

The IMAQ ImageToComplexPlane VI (**Image Processing»Frequency Domain**) and IMAQ ComplexPlaneToImage VI (**Image Processing» Frequency Domain**) allow you to access, process, and update independently the magnitude, phase, real, and imaginary planes of a complex image. You also can convert a complex image to an array and back with the IMAQ ComplexImageToArray VI (**Image Processing» Frequency Domain**) and IMAQ ArrayToComplexImage VI (**Image Processing»Frequency Domain**).

3

Making Grayscale and Color Measurements

This chapter describes how to take measurements from grayscale and color images. You can make inspection decisions based on image statistics, such as the mean intensity level in a region. Based on the image statistics, you can perform many machine vision inspection tasks on grayscale or color images, such as detecting the presence or absence of components, detecting flaws in parts, and comparing a color component with a reference.
Figure 3-1 illustrates the basic steps involved in making grayscale and color measurements.

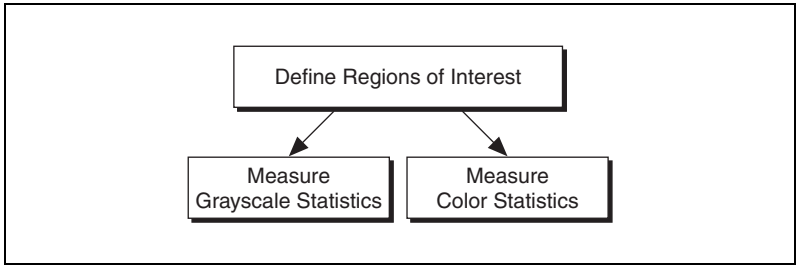


Figure 3-1. Steps for Taking Grayscale and Color Measurements

Define Regions of Interest

A region of interest (ROI) is an area of an image in which you want to focus your image analysis. Table 3-1 describes each of the ROI tools and the manner in which you use them. You can define an ROI interactively, programmatically, or with an image mask.

Table 3-1. Tools Palette Functions

Icon	Tool Name	Function
	Selection Tool	Select an ROI in the image and adjust the position of its control points and contours. Action: Click ROI or control points.
	Point	Select a pixel in the image. Action: Click the position of the pixel.
	Line	Draw a line in the image. Action: Click the initial position and click again at the final position.
	Rectangle	Draw a rectangle or square in the image. Action: Click one corner and drag to the opposite corner.
	Oval	Draw an oval or circle in the image. Action: Click the center position and drag to the required size.
	Polygon	Draw a polygon in the image. Action: Click to place a new vertex and double-click to complete the ROI element.
	Freehand Region	Draw a freehand region in the image. Action: Click the initial position, drag to the required shape and release the mouse button to complete the shape.
	Annulus	Draw an annulus in the image. Action: Click the center position and drag to the required size. Adjust the inner and outer radii, and adjust the start and end angle.
	Zoom	Zoom-in or zoom-out in an image. Action: Click the image to zoom in. Hold down the <Shift> key and click to zoom out.
	Pan	Pan around an image. Action: Click an initial position, drag to the required position and release the mouse button to complete the pan.

Table 3-1. Tools Palette Functions (Continued)

Icon	Tool Name	Function
	Broken Line	Draw a broken line in the image. Action: Click to place a new vertex and double-click to complete the ROI element.
	Freehand Line	Draw a freehand line in the image. Action: Click the initial position, drag to the required shape, and release the mouse button to complete the shape.
	Rotated Rectangle	Draw a rotated rectangle in the image. Action: Click one corner and drag to the opposite corner to create the rectangle. Then click the lines inside the rectangle and drag to adjust the rotation angle.

Hold down the <Shift> key when drawing an ROI if you want to constrain the ROI to the horizontal, vertical, or diagonal axes, when possible. Use the selection tool to position an ROI by its control points or vertices. ROIs are context sensitive, meaning that the cursor actions differ depending on the ROI with which you interact. For example, if you move your cursor over the side of a rectangle, the cursor changes to indicate that you can click and drag the side to resize the rectangle. If you want to draw more than one ROI in an image display environment, hold down the <Ctrl> key while drawing additional ROIs.

Defining Regions Interactively

You can interactively define an ROI using the ROI tools palette in three ways: with the LabVIEW Image Display control, in a separate floating window, or as part of an ROI constructor window.

Defining an ROI in the Image Display Control

To define an ROI within the LabVIEW Image Display control, select an ROI tool from the tools palette, and draw an ROI on your image.

You can configure which ROI tools are present on the control. Complete the following steps to configure the ROI tools palette during design time:

1. Right-click the ROI tools palette and select **Visible Items»ROI Tool Button Visibility**.
2. Deselect the tools you do not want to appear in the ROI tools palette. If you do not want any of the tools to appear, click **All Hidden**.
3. Click **OK** to implement the changes.

To get or set ROIs programmatically, use the property node for the Image Display control.

Defining an ROI in an External Window

The following list describes how you can display the tools palette in an external window and manipulate the palette. You can find all the following VIs on the **Vision Utilities»Region of Interest** palette.

- Use the IMAQ WindToolsShow VI to display the tools window in a floating window.

- Use the IMAQ WindToolsSetup VI to configure the appearance of the tools window.

- Use the IMAQ WindToolsMove VI to move the tools palette.

- Use the IMAQ WindToolsClose VI to close the tools palette.

Note If you want to draw an ROI without displaying the tools palette in an external window, use the IMAQ WindToolsSelect VI (**Vision Utilities»Region of Interest**). This VI allows you to select a contour from the tools palette without opening the palette.

Defining an ROI Using an ROI Constructor

The IMAQ ConstructROI VI enables you to incorporate into your application a LabVIEW modal window in which you select an ROI. This modal window—called an ROI constructor—stops the execution of your application until the user either selects an ROI or clicks **Cancel** to exit the window. The ROI constructor is convenient because it saves you the effort of writing code that behaves similarly. If you want to customize the behavior of the ROI constructor more than is possible with the IMAQ ConstructROI VI, use the Image Display control or the external display window to implement your own ROI constructor.

Complete the following steps to invoke an ROI constructor and define an ROI from within the ROI constructor window:

1. Use the IMAQ ConstructROI VI (**Vision Utilities»Region of Interest**) to display an image and the tools palette in an ROI constructor window, as shown in Figure 3-2.

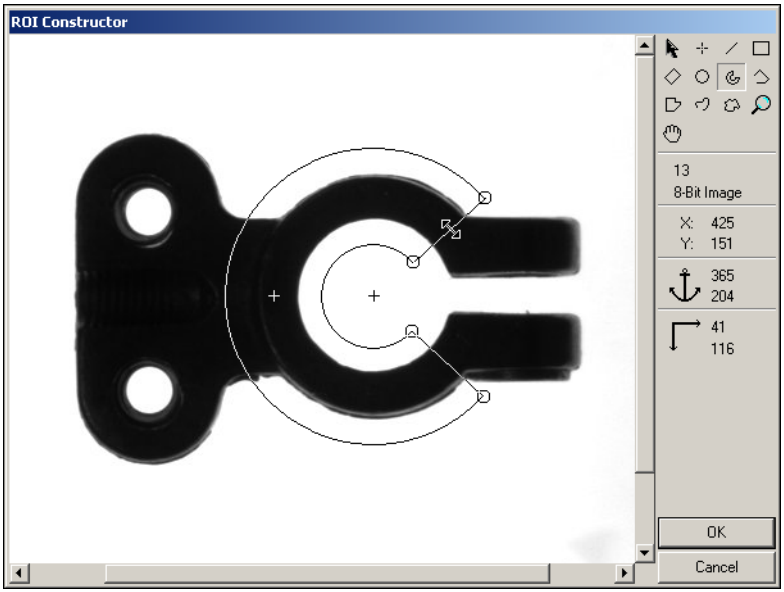


Figure 3-2. ROI Constructor

2. Select an ROI tool from the tools palette.
3. Draw an ROI on your image. Resize and reposition the ROI until it designates the area you want to inspect.
4. Click **OK** to output a descriptor of the region you selected. You can input the ROI descriptor into many analysis and processing functions. You also can convert the ROI descriptor into an image mask, which you can use to process selected regions in the image. Use the IMAQ ROIToMask VI (**Vision Utilities»Region of Interest**) to convert the ROI descriptor into an image mask.

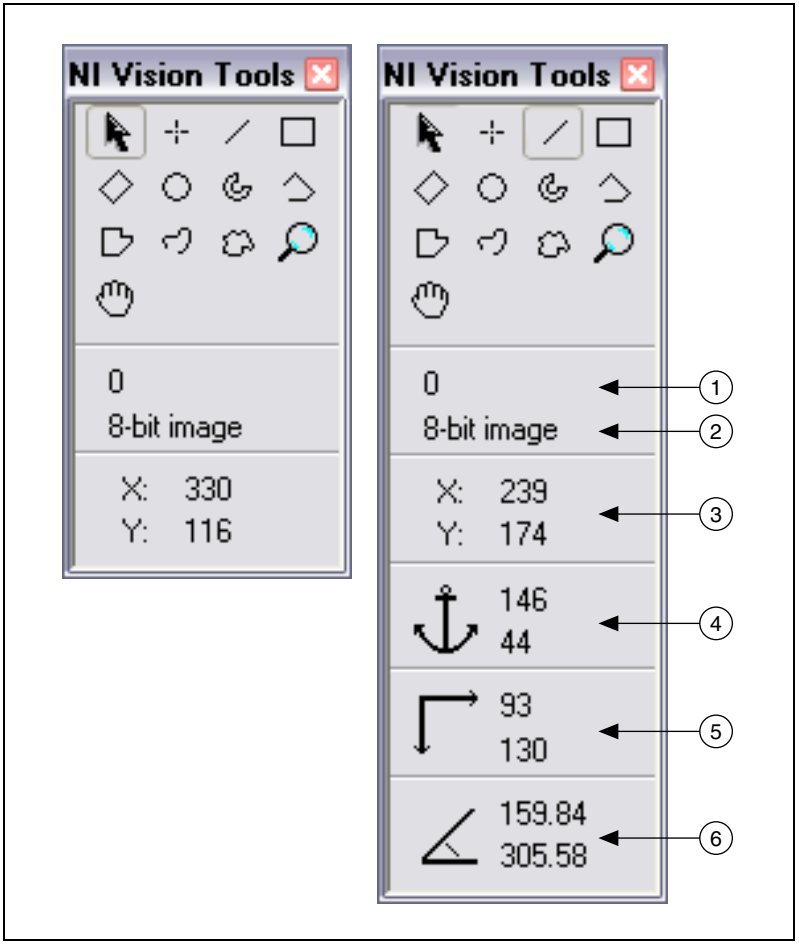
You also can use the IMAQ Select Point VI, IMAQ Select Line VI, and IMAQ Select Rectangle VI to define regions of interest. These three VIs appear in the **Machine Vision»Select Region of Interest** palette. Complete the following steps to use these VIs:

1. Call the VI to display an image in an ROI Constructor window. Only the tools specific to that VI are available for you to use.
2. Select an ROI tool from the tools palette.
3. Draw an ROI on your image. Resize or reposition the ROI until it covers the area you want to process.
4. Click **OK** to output a simple description of the ROI. You can use this description as an input for the VIs on the **Machine Vision»Intensities** palette that measure grayscale intensity:

- IMAQ Light Meter (Point)—Uses the output of IMAQ Select Point
- IMAQ Light Meter (Line)—Uses the output of IMAQ Select Line
- IMAQ Light Meter (Rect)—Uses the output of IMAQ Select Rectangle

Tools Palette Transformation

The tools palette shown in Figure 3-3 is a component of external display windows and ROI constructors. The tools palette automatically transforms from the palette on the left to the palette on the right when you manipulate a region tool in a display window or ROI constructor. The palette on the right displays the characteristics of the ROI you are drawing.



1	Pixel Intensity	4	Anchoring coordinates of an ROI
2	Image Type Indicator (8-bit, 16-bit, Float, Complex, 32-bit RGB, HSL, 64-bit RGB)	5	Size of an active ROI
3	Coordinates of the mouse on the active image window	6	Length and horizontal angle of a line region

Figure 3-3. Tools Palette Tools and Information

Defining Regions Programmatically

When you have an automated application, you may need to define ROIs programmatically. You can programmatically define regions in two ways:

- Specify the elements of an ROI descriptor, which are described as follows:
  - **Global rectangle**—Four integers that describe the bounding rectangle containing all of the shapes in the ROI descriptor.
  - **Contours**—Each of the individual shapes that define an ROI, as follows:
    - **ID**—Specifies if the contour is the external or internal edge of an ROI. If the contour is external, all of the area inside of it is considered part of the ROI. Because external contours are calculated first, internal contours override external contours, giving you the ability to exclude regions inside external contours.
    - **Type**—The shape type of the contour.
    - **Coordinates**—An array containing the coordinates that define the contour.
- Specify regions by providing basic parameters that describe the region you want to define. For example, define a point by providing the x-coordinate and y-coordinate. Define a line by specifying the start and end coordinates. Define a rectangle by specifying the coordinates of the top/left point, bottom/right point, and the rotation angle in thousandths of degrees.

The **Vision Utilities»Region of Interest»Region of Interest Conversion** palette provides VIs to convert simple data types—such as points, lines, rectangles, and annuluses—into an ROI descriptor. Use the following VIs to convert an ROI contour encoded by a simple description to an ROI descriptor for that contour:

- IMAQ Convert Point to ROI—Converts a point specified by its x and y coordinates.

- IMAQ Convert Line to ROI—Converts a line specified by its start and end points.

- IMAQ Convert Rectangle to ROI—Converts a rectangle specified by its top-left and bottom-right points and rotation angle.

- IMAQ Convert Annulus to ROI—Converts an annulus specified by its center point, inner and outer radii, and start and end angles.

- IMAQ Convert Rectangle to ROI (Polygon)—Converts a rectangle specified by its top-left and bottom-right points and rotation angle to an ROI descriptor that uses a polygon to represent the rectangle.

Use the following VIs to convert an ROI contour encoded by an ROI descriptor to a simple description for the contour.

- IMAQ Convert ROI to Point—The output point is specified by its x and y coordinates.

- IMAQ Convert ROI to Line—The output line is specified by its start and end points.

- IMAQ Convert ROI to Rectangle—The output rectangle is specified by its top-left point, bottom-right point, and rotation angle.

- IMAQ Convert ROI to Annulus—The output annulus is specified by its center point, inner and outer radii, and start and end angles.

Defining Regions with Masks

You can define regions to process with image masks. An image mask is an 8-bit image of the same size as or smaller than the image you want to process. Pixels in the image mask determine if the corresponding pixel in the image to process needs to be processed. If a pixel in the image mask has a value different than 0, the corresponding pixel in the image to process is processed. If a pixel in the image mask has a value of 0, the corresponding pixel in the image to process is left unchanged.

Use masks when you need to make intensity measurements on particles in an image. First, threshold your image to make a new binary image. Refer to Chapter 4, *Making Grayscale and Color Measurements*, for information about binary images. You can input the binary image or a labeled version of the binary image as an image mask to the intensity measurement function. If you want to make color comparisons, convert the binary image into an ROI descriptor using the IMAQ MaskToROI VI (**Vision Utilities» Region of Interest**).

Measure Grayscale Statistics

You can measure grayscale statistics in images using light meters or quantitative analysis functions. You can obtain the center of energy for an image with the centroid function.

Use the IMAQ Light Meter (Point) VI (**Machine Vision»Measure Intensities**) to measure the light intensity at a point in the image. Use the IMAQ Light Meter (Line) VI (**Machine Vision»Measure Intensities**) to get the pixel value statistics along a line in the image: mean intensity, standard deviation, minimum intensity, and maximum intensity. Use the IMAQ Light Meter (Rectangle) VI (**Machine Vision»Measure Intensities**) to get the pixel value statistics within a rectangular region in an image.

Use the IMAQ Quantify VI (**Image Processing»Analysis**) to obtain the following statistics about the entire image or individual regions in the image: mean intensity, standard deviation, minimum intensity, maximum intensity, area, and the percentage of the image that you analyzed. You can specify regions in the image with a labeled image mask. A labeled image mask is a binary image that has been processed so that each region in the image mask has a unique intensity value. Use the IMAQ Label VI (**Image Processing»Processing**) to label your image mask.

Use the IMAQ Centroid VI (**Image Processing»Analysis**) to compute the energy center of the image or of a region within an image.

Measure Color Statistics

Most image processing and analysis functions apply to 8-bit and 16-bit images. However, you can analyze and process individual components of a color image.

Using the IMAQ ExtractColorPlanes VI (**Vision Utilities»Color Utilities**), you can break down a color image into various sets of primary components, such as RGB (Red, Green, and Blue), HSI (Hue, Saturation, and Intensity), HSL (Hue, Saturation, and Luminance), or HSV (Hue, Saturation, and Value). Each component becomes an 8-bit or 16-bit image that you can process like any other grayscale image. Using the

IMAQ ReplaceColorPlane VI (**Vision Utilities»Color Utilities**), you can reassemble a color image from a set of three 8-bit or 16-bit images, where each image becomes one of the three primary components. Figure 3-4 and

Figure 3-5 illustrate how 32-bit and 64-bit color images break down into their three primary components.

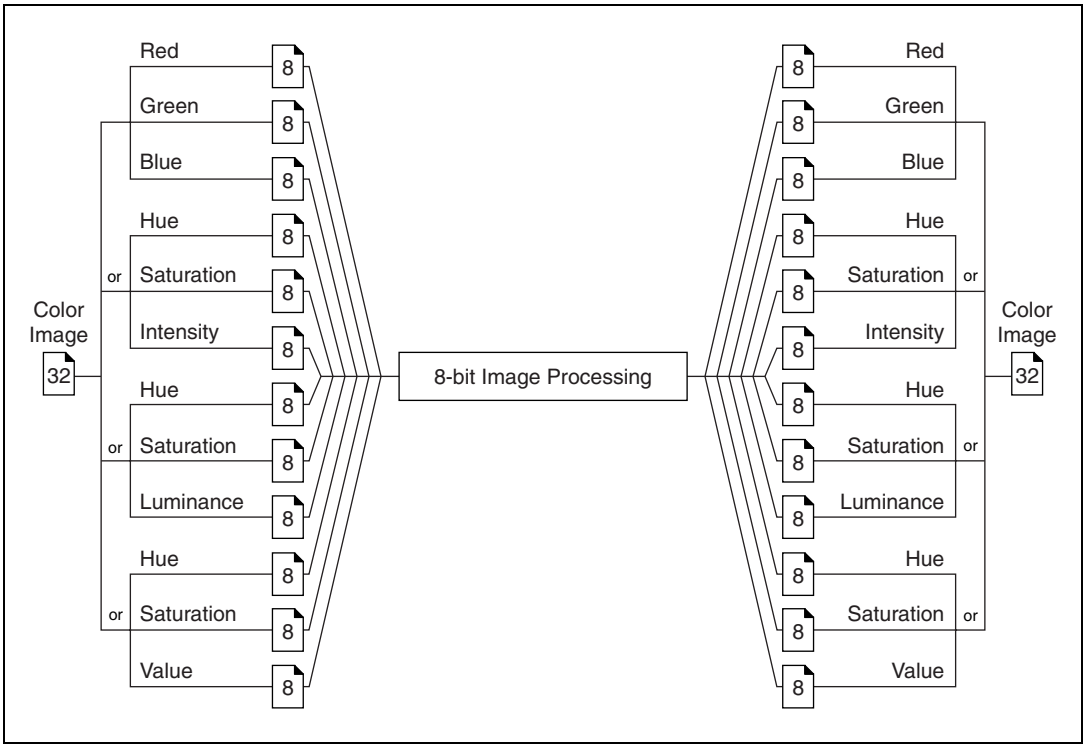


Figure 3-4. Primary Components of a 32-Bit Color Image

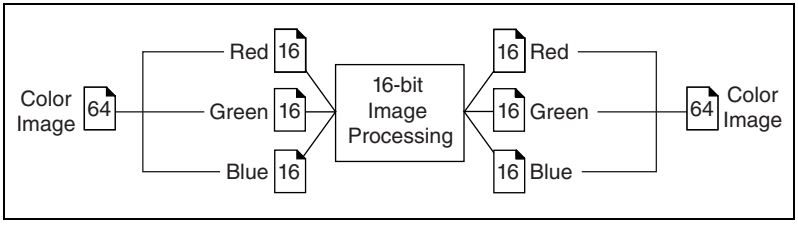
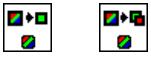


Figure 3-5. Primary Components of a 64-Bit Color Image



Use the IMAQ ExtractSingleColorPlane VI or the IMAQ ExtractColorPlanes VI (**Vision Utilities»Color Utilities**) to extract the red, green, blue, hue saturation, intensity, luminance, or value plane of a color image into an 8-bit image.

A color pixel encoded as an unsigned 32-bit integer control can be decomposed into its individual components using the IMAQ IntegerToColorValue VI. You can convert a pixel value represented by its RGB components into the equivalent components for another color model using the IMAQ RGBToColor 2 VI. You can convert components in any other color mode to RGB by using the IMAQ ColorToRGB VI. These three VIs appear in the **Vision Utilities»Color Utilities** palette.

Comparing Colors

You can use the color matching capability of NI Vision to compare or evaluate the color content of an image or regions in an image. Complete the following steps to compare colors using color matching:

1. Select an image containing the color information that you want to use as a reference. The color information can consist of multiple colors.
2. Use the entire image or regions in the image to learn the color information using the IMAQ ColorLearn VI (**Image Processing» Color Processing**), which outputs a color spectrum that contains a compact description of the color information that you learned. Refer to Chapter 15, *Color Inspection*, of the *NI Vision Concepts Manual* for more information. Use the color spectrum to represent the learned color information for all subsequent matching operations.
3. Define an image or multiple regions in an image as the inspection or comparison area.
4. Use the IMAQ ColorMatch VI (**Image Processing»Color Processing**) to compare the learned color information to the color information in the inspection regions. This VI returns a score that indicates the closeness of match. You can specify a **Minimum Match Score**, which indicates if there is a match between the input color information and the color in each specified region in the image.
5. Use the color matching score as a measure of similarity between the reference color information and the color information in the image regions being compared.

Learning Color Information

When learning color information, choose the color information carefully by specifying an image or regions in an image that contain the color or color set that you want to learn, selecting how detailed you want the color information to be learned, and choosing colors that you want to ignore during matching.

Specifying the Color Information to Learn

Because color matching uses only color information to measure similarity, the image or regions in the image representing the object should contain only the significant colors that represent the object, as shown in Figure 3-6a. Figure 3-6b illustrates an unacceptable region containing background colors.

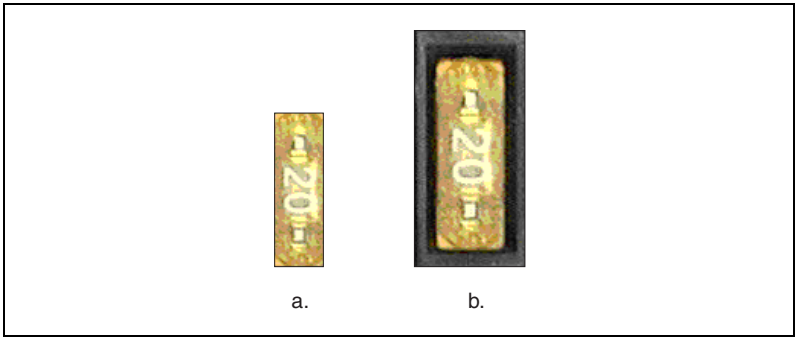


Figure 3-6. Template Color Information

The following sections explain when to learn the color information associated with an entire image, a region in an image, or multiple regions in an image.

Using the Entire Image

You can use an entire image to learn the color spectrum that represents the entire color distribution of the image. In a fabric identification application, for example, an entire image can specify the color information associated with a certain fabric type, as shown in Figure 3-7.

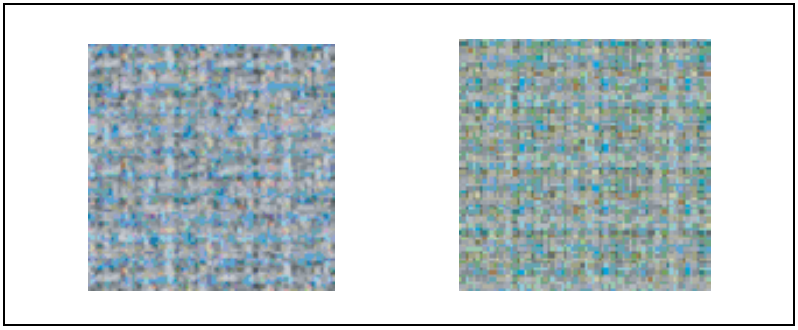


Figure 3-7. Using the Entire Image to Learn Color Distribution

Using a Region in the Image

You can select a region in the image to provide the color information for comparison. A region is helpful for pulling out the useful color information in an image. Figure 3-8 shows an example of using a region that contains the color information that is important for the application.

Figure 3-8. Using a Single Region to Learn Color Distribution

Using Multiple Regions in the Image

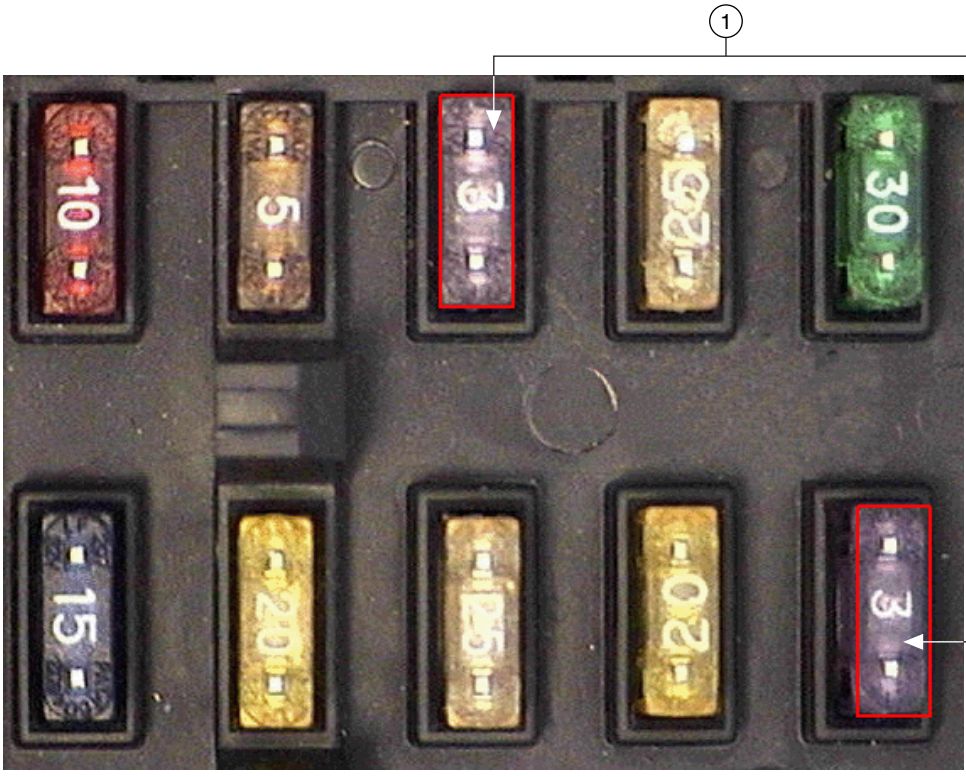
The interaction of light with an object surface creates the observed color of that object. The color of a surface depends on the directions of illumination and the direction from which the surface is observed. Two identical objects may have different appearances because of a difference in positioning or a change in the lighting conditions.

Figure 3-9 shows how light reflects differently off of the 3D surfaces of the fuses, resulting in slightly different colors for identical fuses. Compare the 3-amp fuse in the upper row with the 3-amp fuse in the lower row. The difference in light reflection results in different color spectrums for identical fuses.

If you learn the color spectrum by drawing a region of interest inside the 3-amp fuse in the upper row and then do a color matching for the 3-amp fuse in the upper row, you get a very high match score close to 1000. However, the match score for the 3-amp fuse in the lower row is quite low at around 500. This problem could cause a mismatch for the color matching in a fuse box inspection process.

The color learning algorithm of NI Vision uses a clustering process to find the representative colors from the color information specified by one or multiple regions in the image. To create a representative color spectrum for all 3-amp fuses in the learning phase, draw an ROI around the three amp fuse in the upper row, hold down the <Ctrl> key, and draw another ROI around the 3-amp fuse in the lower row. The new color spectrum represents

3-amp fuses much better and results in high match scores—near 800—for both 3-amp fuses. Use as many samples as you want in an image to learn the representative color spectrum for a specified template.



1 Regions Used to Learn Color Information

Figure 3-9. Using Multiple Regions to Learn Color Distribution

Choosing a Color Representation Sensitivity

When you learn a color, you need to specify the sensitivity required to specify the color information. An image containing a few, well-separated colors in the color space requires a lower sensitivity to describe the color than an image that contains colors that are close to one another in the color space. Use the **Color Sensitivity** control of the IMAQ ColorLearn VI to specify the granularity you want to use to represent the colors. Refer to the *Color Sensitivity* section of Chapter 5, *Performing Machine Vision Tasks*, for more information about color sensitivity.

Ignoring Learned Colors

Ignore certain color components in color matching by replacing the corresponding component in the input color spectrum array to –1. For example, by replacing the last component in the color spectrum with –1, the white color is ignored during the color matching process. By replacing the second to last component in the color spectrum, the black color is ignored during the color matching process.

To ignore other color components in color matching, determine the index to the color spectrum by locating the corresponding bins in the color wheel, where each bin corresponds to a component in the color spectrum array. Ignoring certain colors, such as the background color, results in a more accurate color matching score. Ignoring the background color also provides more flexibility when defining the regions of interest in the color matching process. Ignoring certain colors, such as the white color created by glare on a metallic surface, also improves the accuracy of the color matching. Experiment learning the color information about different parts of the images to determine which colors to ignore. Refer to Chapter 15, *Color Inspection*, of the *NI Vision Concepts Manual* for more information about the color wheel and color bins.

4

Performing Particle Analysis

This chapter describes how to perform particle analysis on your images. Use particle analysis to find statistical information—such as the area, number, location, and presence of particles. With this information, you can perform many machine vision inspection tasks, such as detecting flaws on silicon wafers or detecting soldering defects on electronic boards. Examples of how particle analysis can help you perform web inspection tasks include locating structural defects on wood planks or detecting cracks on plastic sheets.

Figure 4-1 illustrates the steps involved in performing particle analysis.

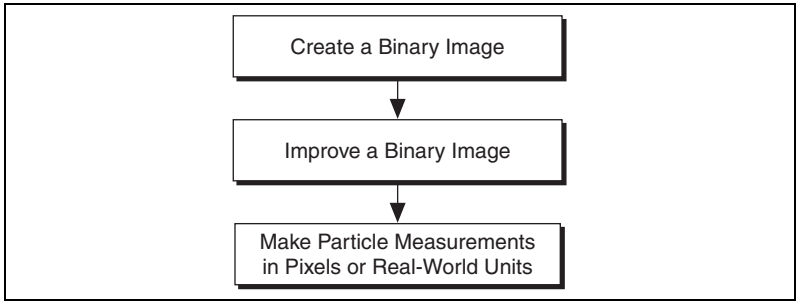


Figure 4-1. Steps for Performing Particle Analysis

Create a Binary Image

Threshold your grayscale or color image to create a binary image. Creating a binary image separates the objects that you want to inspect from the background. The threshold operation sets the background pixels to 0 in the binary image, while setting the object pixels to a non-zero value. Object pixels have a value of 1 by default, but you can set the object pixels to have any value you choose.

You can use different techniques to threshold your image. If all the objects of interest in your grayscale image fall within a continuous range of intensities and you can specify this threshold range manually, use the IMAQ Threshold VI (**Image Processing»Processing**) to threshold your image. If all the objects in your grayscale image are either brighter or

darker than your background, you can use one of the automatic thresholding techniques in NI Vision. Complete the following steps to use one of the automatic thresholding techniques:

1. Use the IMAQ AutoBThreshold VI (**Image Processing»Processing**) to select the thresholding technique that automatically determines the optimal threshold range.

2. Connect the **Threshold Data** output to the IMAQ MultiThreshold VI (**Image Processing»Processing**), or use the **Lookup Table** output to apply a lookup table to the image using the IMAQ UserLookup VI (**Image Processing»Processing**).

If your grayscale image contains objects that have multiple discontinuous grayscale values, use the IMAQ MultiThreshold VI (**Image Processing» Processing**).

If your grayscale image contains objects whose grayscale values vary within the image due to effects such as light drift, use the IMAQ LocalThreshold VI (**Image Processing» Processing**).

If you need to threshold a grayscale image that has nonuniform lighting conditions, such as those resulting from a strong illumination gradient or shadows, use the IMAQ LocalThreshold VI (**Image Processing» Processing**). You need to define a pixel window that specifies which neighboring pixels are considered in the statistical calculation. The default window size is 32×32. However, the window size should be approximately the size of the smallest object you want to separate from the background. You also need to specify the local thresholding algorithm to use. The local thresholding algorithm options are the Niblack or background correction algorithm. Refer to Chapter 8, *Image Segmentation*, of the *NI Vision Concepts Manual* for more information about local thresholding.

Automatic thresholding techniques offer more flexibility than simple thresholds based on fixed ranges. Because automatic thresholding techniques determine the threshold level according to the image *histogram*, the operation is less affected by changes in the overall brightness and contrast of the image than a fixed threshold. Because these techniques are more resistant to changes in lighting, they are well suited for automated inspection tasks.

If you need to threshold a color image, use the IMAQ ColorThreshold VI (**Image Processing»Color Processing**). You must specify threshold ranges for each of the color planes—either Red, Green, and Blue or Hue, Saturation, and Luminance). The binary image resulting from a color threshold is an 8-bit binary image.

Improve the Binary Image

After you threshold your image, you may want to improve the resulting binary image with binary morphological functions. You can use primary *binary morphology* or advanced binary morphology to remove unwanted particles, separate connected particles, or improve the shape of particles. Primary morphology functions work on the image as a whole by processing pixels individually. Advanced morphology operations are built upon the primary morphological operators and work on particles as a whole as opposed to individual pixels. Refer to Chapter 9, *Binary Morphology*, of the *NI Vision Concepts Manual* for lists of which morphology functions are primary and which are advanced.

The advanced morphology functions require that you specify the type of *connectivity* to use. Connectivity specifies how NI Vision determines if two adjacent pixels belong to the same particle. Use *connectivity-4* when you want NI Vision to consider pixels to be part of the same particle only when the pixels touch along an adjacent edge. Use *connectivity-8* when you want NI Vision to consider pixels to be part of the same particle even if the pixels touch only at a corner. Refer to Chapter 9, *Binary Morphology*, of the *NI Vision Concepts Manual* for more information about connectivity.

Note Use the same type of connectivity throughout your application.

Removing Unwanted Particles

Use the IMAQ RejectBorder VI (**Image Processing»Morphology**) to remove particles that touch the border of the image. Reject particles on the border of the image when you suspect that the information about those particles is incomplete.

Use the IMAQ RemoveParticle VI (**Image Processing»Morphology**) to remove large or small particles that do not interest you. You also can use the Erode, Open, and POpen functions in the IMAQ Morphology VI (**Image Processing»Morphology**) to remove small particles. Unlike the IMAQ RemoveParticle VI, these three functions alter the size and shape of the remaining particles.

Use the hit-miss function of the IMAQ Morphology VI to locate particular configurations of pixels, which you define with a structuring element. Depending on the configuration of the structuring element, the hit-miss function can locate single isolated pixels, cross-shape or longitudinal patterns, right angles along the edges of particles, and other user-specified shapes. Refer to Chapter 9, *Binary Morphology*, of the *NI Vision Concepts Manual* for more information about structuring elements.

If you know enough about the shape features of the particles you want to keep or remove, use the IMAQ Particle Filter 2 VI (**Image Processing» Morphology**) to filter out particles that do not interest you.

Separating Touching Particles

Use watershed transform and binary morphology to separate touching particles in images.

Using Watershed Transform

Use the IMAQ Danielsson VI (**Image Processing»Morphology**) to transform the binary image into a grayscale distance map in which each particle pixel is assigned a gray-level value equal to its shortest Euclidean distance from the particle border. Apply the IMAQ Watershed Transform VI (**Image Processing»Processing**) to the distance map to find the watershed separation lines. Use the IMAQ Mask VI (**Image Processing»Operators**) to superimpose the watershed lines on the original image.

Refer to Chapter 8, *Image Segmentation*, of the *NI Vision Concepts Manual* for more information about segmenting images using a watershed transform.

Using Binary Morphology

Use the IMAQ Separation VI (**Image Processing»Morphology**) or apply an erosion or an open function with the IMAQ Morphology VI (**Image Processing»Morphology**) to separate touching objects. IMAQ Separation is an advanced VI that separates particles without modifying their shapes. However, erosion and open operations alter the shape of all the particles.

Note A separation is a time-intensive operation compared to an erosion, watershed transform, or open operation. Consider using one of the methods other than separation if speed is an issue in your application.

Improving Particle Shapes

Use the IMAQ FillHoles VI (**Image Processing»Morphology**) to fill holes in the particles. Use the IMAQ Morphology VI (**Image Processing» Morphology**) to perform a variety of operations on the particles. You can use the Open, Close, POpen, PClose, and AutoM operations to smooth the boundaries of the particles. Open and POpen smooth the boundaries of the particle by removing small isthmuses while Close widens the isthmuses. Close and PClose fill small holes in the particle. AutoM removes isthmuses and fills holes. Refer to Chapter 9, *Binary Morphology*, of the *NI Vision Concepts Manual* for more information about these operations.

Make Particle Measurements

After you create a binary image and improve it, you can make particle measurements. NI Vision can return the measurements in uncalibrated pixels or calibrated real-world units. With these measurements you can determine the location of particles and their shape features. Use the following VIs to perform particle measurements:

- IMAQ Particle Analysis Report (**Image Processing»Analysis**)—This VI returns the number of particles in an image and an array of reports containing 11 of the most commonly used measurements, including the particle area, bounding rectangle, and center of mass.

- IMAQ Particle Analysis (**Image Processing»Analysis**)—This VI returns an array containing your choice of up to 80 particle measurements.

  Refer to Chapter 10, *Particle Measurements*, of the *NI Vision Concepts Manual* for a list of all the measurements that IMAQ Particle Analysis can return.

5

Performing Machine Vision Tasks

This chapter describes how to perform many common machine vision inspection tasks. The most common inspection tasks are detecting the presence or absence of parts in an image and measuring the dimensions of parts to determine if they meet specifications.

Measurements are based on characteristic features of the object represented in the image. Image processing algorithms traditionally classify the type of information contained in an image as edges, surfaces and textures, or patterns. Different types of machine vision algorithms leverage and extract one or more types of information.

Edge detectors and derivative techniques—such as rakes, concentric rakes, and spokes—use edges represented in the image. They locate, with high accuracy, the position of the edge of an object. You can use edge detection to make such measurements as the width of the part, which is a technique called *clamping*. You also can combine multiple edge locations to compute intersection points, projections, circles, or ellipse fits.

Pattern matching algorithms use edges and patterns. Pattern matching can locate, with very high accuracy, the position of fiducials or characteristic features of the part under inspection. You can combine those locations to compute lengths, angles, and other object measurements.

The robustness of your measurement relies on the stability of your image acquisition conditions. Sensor resolution, lighting, optics, vibration control, part fixture, and general environment are key components of the imaging setup. All elements of the image acquisition chain directly affect the accuracy of the measurements.

Figure 5-1 illustrates the basic steps involved in performing machine vision.

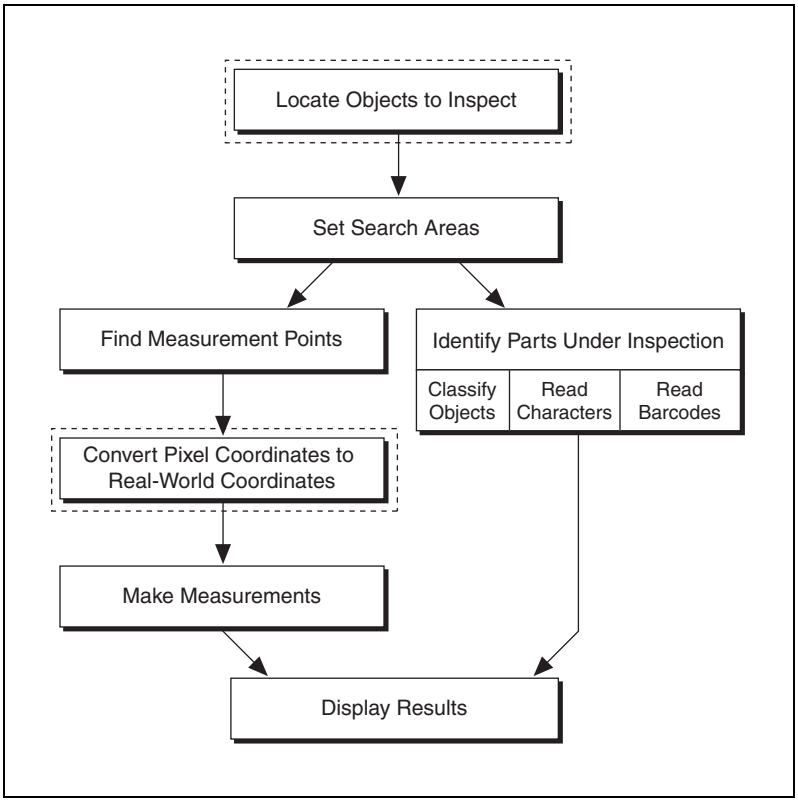


Figure 5-1. Steps for Performing Machine Vision

Note Diagram items enclosed with dashed lines are optional steps.

Locate Objects to Inspect

In a typical machine vision application, you extract measurements from ROIs rather than the entire image. To use this technique, the parts of the object you are interested in must always appear inside the ROIs you define.

If the object under inspection is always at the same location and orientation in the images you need to process, defining ROIs is straightforward. Refer to the *Set Search Areas* section of this chapter for information about selecting an ROI.

Often, the object under inspection appears shifted or rotated in the image you need to process, relative to the reference image in which you located the object. When this occurs, the ROIs need to shift and rotate with the parts of the object in which you are interested. For the ROIs to move with the object, you need to define a reference coordinate system relative to the object in the reference image. During the measurement process, the coordinate system moves with the object when the object appears shifted and rotated in the image you need to process. This coordinate system is referred to as the measurement coordinate system. The measurement VIs automatically move the ROIs to the correct position using the position of the measurement coordinate system relative to the reference coordinate system. Refer to Chapter 14, *Dimensional Measurements*, of the *NI Vision Concepts Manual* for information about coordinate systems.

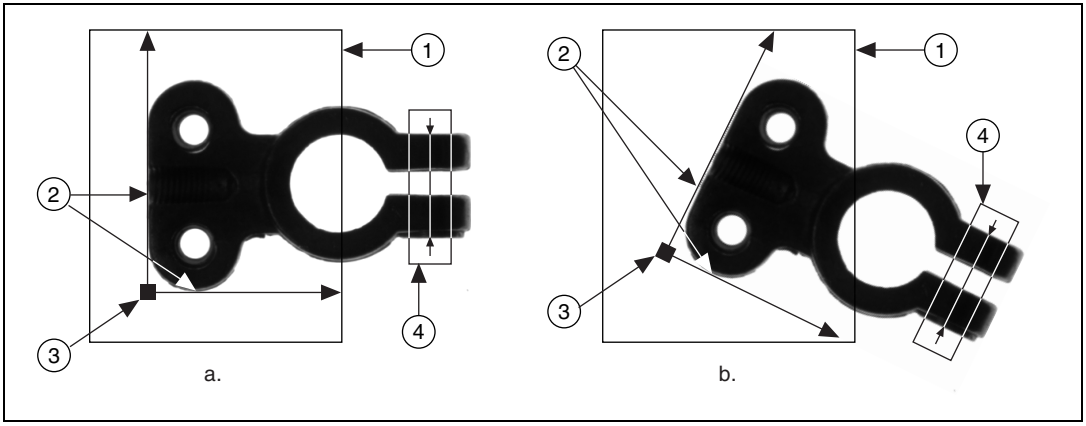
You can build a coordinate transformation using edge detection or pattern matching. The output of the edge detection and pattern matching VIs that build a coordinate transformation are the origin, angle, and axes direction of the coordinate system. Some machine vision VIs take this output and adjust the regions of inspection automatically. You also can use these outputs to move the regions of inspection relative to the object programmatically.

Using Edge Detection to Build a Coordinate Transformation

You can build a coordinate transformation using two edge detection techniques. Use the IMAQ Find CoordSys (Rect) VI (**Machine Vision» Coordinate Systems**) to define a reference coordinate system using one rectangular region. Use the IMAQ Find CoordSys (2 Rects) VI (**Machine Vision»Coordinate Systems**) to define a reference coordinate system using two independent rectangular regions. Complete the following steps to build a coordinate transformation using edge detection:

Note To use these techniques, the object cannot rotate more than ±65° in the image.

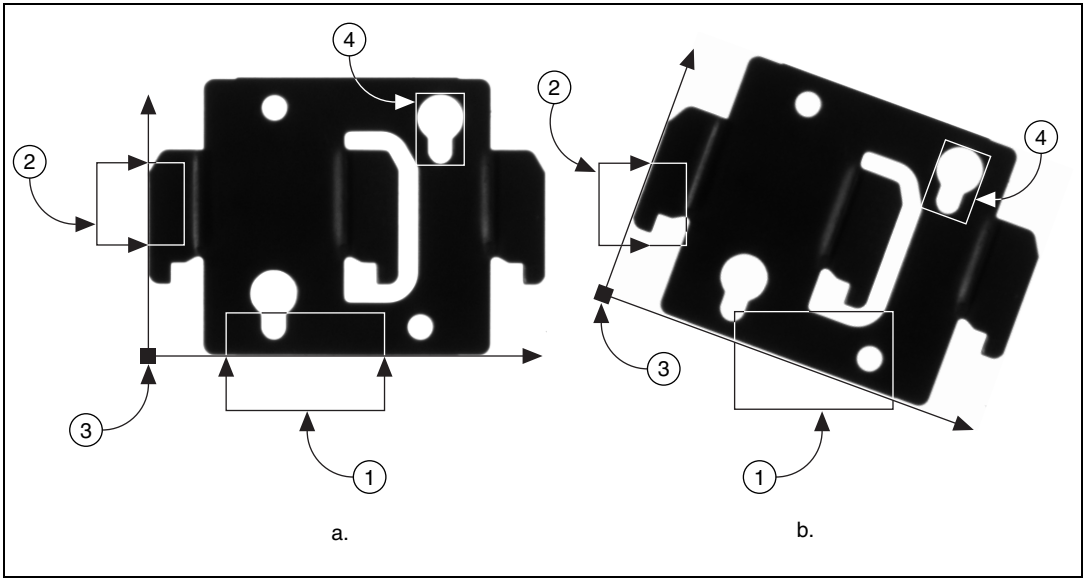
1. Specify one or two rectangular regions.
   a. If you use IMAQ Find CoordSys (Rect), specify one rectangular ROI that includes part of two straight, nonparallel boundaries of the object, as shown in Figure 5-2. This rectangular region must be large enough to include these boundaries in all the images you want to inspect.



1 Search Area for the Coordinate System	3 Origin of the Coordinate System
2 Object Edges	4 Measurement Area

Figure 5-2. Coordinate Systems of a Reference Image and Inspection Image

b. If you use IMAQ Find CoordSys (2 Rects), specify two rectangular ROIs, each containing one separate, straight boundary of the object, as shown in Figure 5-3. The boundaries cannot be parallel. The regions must be large enough to include the boundaries in all the images you want to inspect.



1 Primary Search Area	3 Origin of the Coordinate System
2 Secondary Search Area	4 Measurement Area

Figure 5-3. Locating Coordinate System Axes with Two Search Areas

2. Choose the parameters you need to locate the edges on the object.
3. Choose the coordinate system axis direction.
4. Choose the results that you want to overlay onto the image.
5. Choose the mode for the VI. To build a coordinate transformation for the first time, set mode to **Find Reference**. To update the coordinate transformation in subsequent images, set this mode to **Update Reference**.

Using Pattern Matching to Build a Coordinate Transformation

You can build a coordinate transformation using pattern matching. Use the IMAQ Find CoordSys (Pattern) 2 VI (**Machine Vision»Coordinate Systems**) to define a reference coordinate system based on the location of a reference feature. Use this technique when the object under inspection does not have straight, distinct edges. Complete the following steps to build a coordinate transformation using pattern matching:

Note The object can rotate 360° in the image using this technique if you use rotation-invariant pattern matching.

1. Define a template that represents the part of the object that you want to use as a reference feature. Refer to the *Find Measurement Points* section for information about defining a template.
2. Define a rectangular search area in which you expect to find the template.
3. Choose the **Match Mode**. Select **Rotation Invariant** when you expect your template to appear rotated in the inspection images. Otherwise, select **Shift Invariant**.
4. Choose the results that you want to overlay onto the image.
5. Choose the mode for the VI. To build a coordinate transformation for the first time, set mode to **Find Reference**. To update the coordinate transformation in subsequent images, set this mode to **Update Reference**.

Choosing a Method to Build the Coordinate Transformation

The flowchart in Figure 5-4 guides you through choosing the best method for building a coordinate transformation for your application.

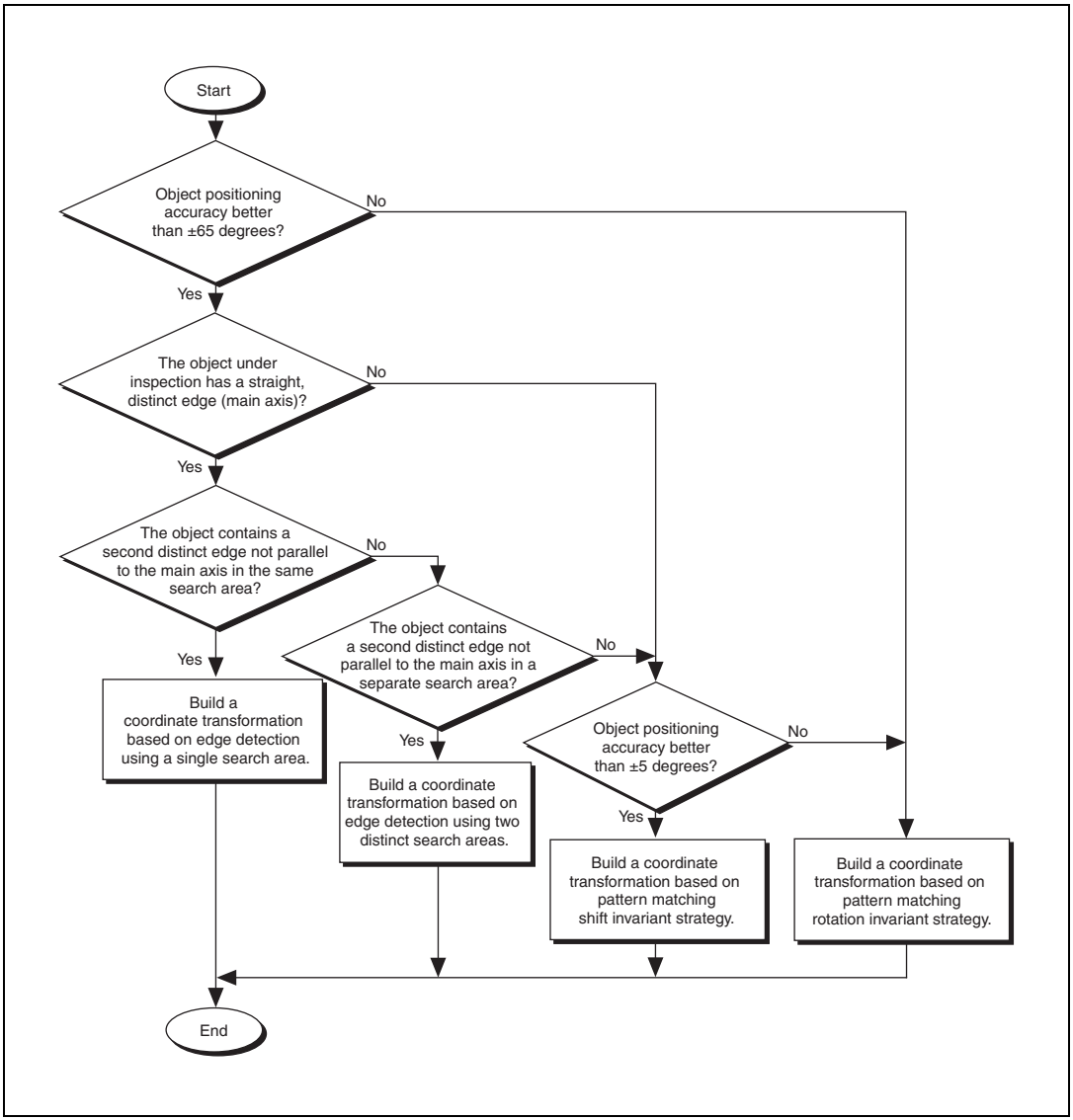

Start
Object positioning accuracy better than ±65 degrees?
No
Yes
The object under inspection has a straight, distinct edge (main axis)?
No
Yes
The object contains a second distinct edge not parallel to the main axis in the same search area?
No
Yes
The object contains a second distinct edge not parallel to the main axis in a separate search area?
No
Yes
Object positioning accuracy better than ±5 degrees?
No
Yes
Build a coordinate transformation based on edge detection using a single search area.
Build a coordinate transformation based on edge detection using two distinct search areas.
Build a coordinate transformation based on pattern matching shift invariant strategy.
Build a coordinate transformation based on pattern matching rotation invariant strategy.
End

Figure 5-4. Building a Coordinate Transformation

Set Search Areas

Select ROIs in your images to limit the areas in which you perform your processing and inspection. You can define ROIs interactively or programmatically.

Defining Regions Interactively

Use the techniques described in Chapter 4, *Performing Particle Analysis*, to select an ROI. The following table shows which ROI to use with a given measurement VI.

ROI	Measurement VI
Rotated rectangle	IMAQ Find Pattern (**Machine Vision»Find Patterns**) IMAQ Clamp Horizontal Max (**Machine Vision»Measure Distances**) IMAQ Clamp Horizontal Min (**Machine Vision»Measure Distances**) IMAQ Clamp Vertical Max (**Machine Vision»Measure Distances**) IMAQ Clamp Vertical Min (**Machine Vision»Measure Distances**) IMAQ Find Horizontal Edge (**Machine Vision»Locate Edges**) IMAQ Find Vertical Edge (**Machine Vision»Locate Edges**)
Annulus	IMAQ Find Circular Edge (**Machine Vision»Locate Edges**) IMAQ Find Concentric Edge (**Machine Vision»Locate Edges**)

Defining Regions Programmatically

When you design an automated application, you need to define ROIs programmatically. You can programmatically define regions in two ways:

- Specify the elements of the ROI descriptor.
- Specify regions by providing basic parameters that describe the region you want to define. You can specify a rotated rectangle by providing the coordinates of the top, left point, and the bottom, right point, and the rotation angle. You can specify an annulus by providing the coordinates of the center, inner radius, outer radius, start angle, and end angle. You can specify a point by setting its x-coordinates and y-coordinates. You can specify a line by setting the coordinates of the start and end points.

Refer to Chapter 4, *Performing Particle Analysis*, for more information about defining ROIs.

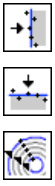
Find Measurement Points

After you set regions of inspection, you can locate points within those regions on which you can base measurements. You can locate measurement points using edge detection, pattern matching, color pattern matching, and color location.

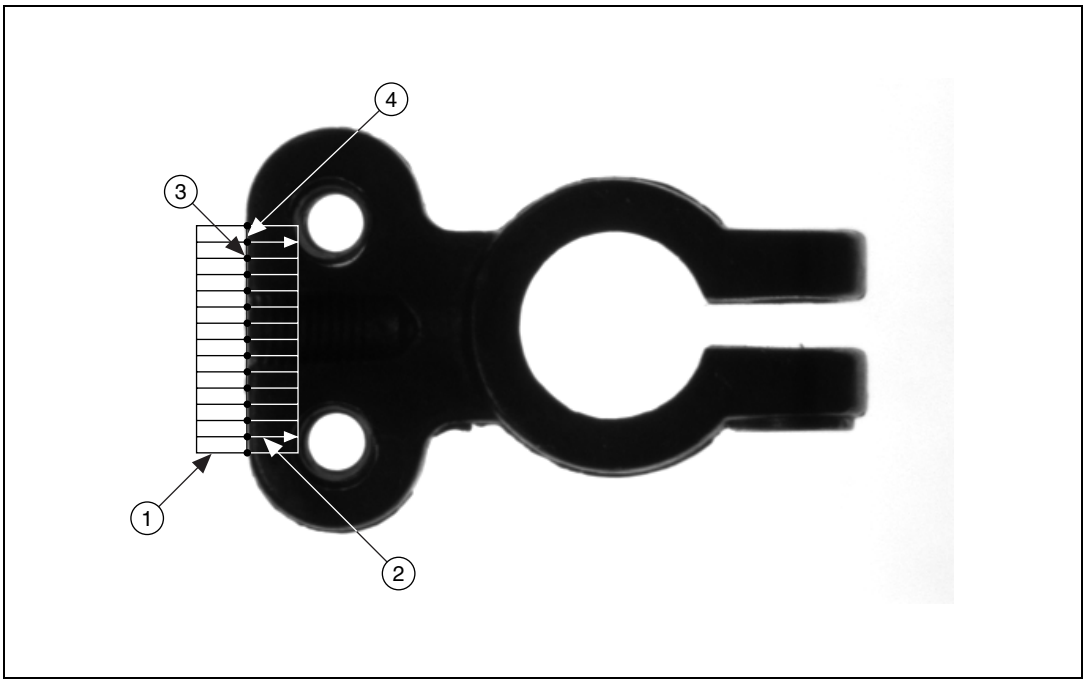
Finding Features Using Edge Detection

Discontinuities in an image typically represent abrupt changes in pixel intensity values, which characterize the boundaries of objects. Use the edge detection tools to identify and locate sharp discontinuities in an image.

Finding Lines or Circles



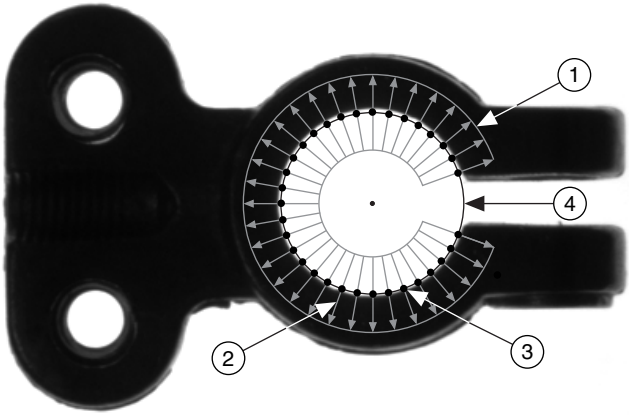
If you want to find points along the edge of an object and find a line describing the edge, use the IMAQ Find Vertical Edge, IMAQ Find Horizontal Edge, and IMAQ Find Concentric Edge VIs (**Machine Vision»Locate Edges**). IMAQ Find Vertical Edge and IMAQ Find Horizontal Edge find edges based on rectangular search areas, as shown in Figure 5-5. IMAQ Find Concentric Edge finds edges based on annular search areas.



1 Search Region 2 Search Lines 3 Detected Edge Points 4 Line Fit to Edge Points

Figure 5-5. Finding a Straight Feature

If you want to find points along a circular edge and find the circle that best fits the edge, as shown in Figure 5-6, use the IMAQ Find Circular Edge VI (**Machine Vision»Locate Edges**).



1	Annular Search Region	3	Detected Edge Points
2	Search Lines	4	Circle Fit To Edge Points

Figure 5-6. Finding a Circular Feature

IMAQ Find Vertical Edge, IMAQ Find Horizontal Edge, and IMAQ Find Concentric Edge locate the intersection points between a set of search lines within the search region and the edge of an object. Specify the separation between the lines that the VIs use to detect edges. The VIs determine the intersection points based on their contrast, width, and steepness. The software calculates a best-fit line with outliers rejected or a best-fit circle through the points it found. The VIs return the coordinates of the edges found.

Finding Edge Points Along One Search Contour

Use the IMAQ Simple Edge VI (**Machine Vision»Caliper**) and the IMAQ Edge Tool VI (**Machine Vision»Caliper**) to find edge points along a contour. You can find the first edge, last edge, or all edges along the contour. Use IMAQ Simple Edge when your image contains little noise and the object and background are clearly differentiated. Otherwise, use IMAQ Edge Tool.

IMAQ Simple Edge and IMAQ Edge Tool require you to input the coordinates of the points along the search contour. Use the IMAQ ROIProfile VI (**Image Processing»Analysis**) to obtain the coordinates from the ROI descriptor of the contour. If you have a straight line, use the IMAQ GetPointsOnLine VI (**Machine Vision»Analytic**

Geometry) to obtain the points along the line instead of using an ROI descriptor.

IMAQ ROIProfile and IMAQ GetPointsOnLine determine the edge points based on their contrast and slope. You can specify if you want to find the edge points using subpixel accuracy.

Finding Edge Points Along Multiple Search Contours

Use the IMAQ Rake, IMAQ Spoke, and IMAQ Concentric Rake VIs (**Machine Vision»Caliper**) to find edge points along multiple search contours. These functions behave like the IMAQ Edge Tool VI, but they find edges on multiple contours. Pass in an ROI descriptor to define the search region for these VIs.

IMAQ Rake works on a rectangular search region. The search lines are drawn parallel to the orientation of the rectangle. Control the number of search lines in the region by specifying the distance, in pixels, between each line. Specify the search direction as left to right or right to left for a horizontally oriented rectangle. Specify the search direction as top to bottom or bottom to top for a vertically oriented rectangle.

IMAQ Spoke works on an annular search region, scanning the search lines that are drawn from the center of the region to the outer boundary and that fall within the search area. Control the number of lines in the region by specifying the angle, in degrees, between each line. Specify the search direction as either going from the center outward or from the outer boundary to the center.

IMAQ Concentric Rake works on an annular search region. The concentric rake is an adaptation of the rake to an annular region. Edge detection is performed along search lines that occur in the search region and that are concentric to the outer circular boundary. Control the number of concentric search lines that are used for the edge detection by specifying the radial distance between the concentric lines in pixels. Specify the direction of the search as either clockwise or counterclockwise.

Finding Points Using Pattern Matching

The pattern matching algorithms in NI Vision measure the similarity between an idealized representation of a feature, called a template, and the feature that may be present in an image. A feature is a specific pattern of pixels in an image. Pattern matching returns the location of the center of the template and the template orientation. Complete the following generalized steps to find features in an image using pattern matching:

1. Define a reference or fiducial pattern to use as a template image.
2. Use the reference pattern to train the pattern matching algorithm with IMAQ Learn Pattern 2.
3. Define an image or an area of an image as the search area. A small search area can reduce the time to find the features.
4. Set the tolerances and parameters to specify how the algorithm operates at run time using IMAQ Setup Match Pattern 2.
5. Test the search algorithm on test images using IMAQ Match Pattern 2.
6. Verify the results using a ranking method.

Defining and Creating Effective Template Images

The selection of an effective template image plays a critical part in obtaining good results. Because the template image represents the pattern that you want to find, make sure that all the important and unique characteristics of the pattern are well defined in the image.

Several factors are critical in creating a template image. These critical factors include symmetry, feature detail, positional information, and background information.

Symmetry

A rotationally symmetric template, as shown in Figure 5-7a is less sensitive to changes in rotation than one that is rotationally asymmetric.
A rotationally symmetric template provides good positioning information but no orientation information.

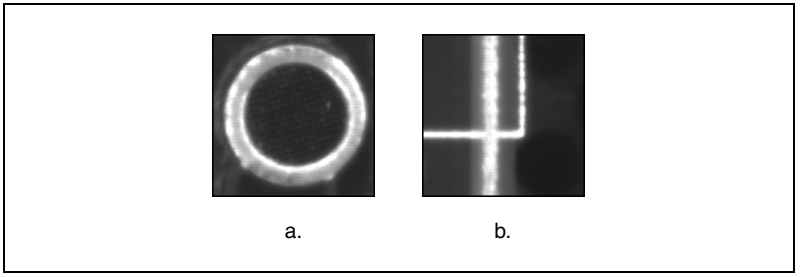


Figure 5-7. Symmetry

Feature Detail

A template with relatively coarse features is less sensitive to variations in size and rotation than a model with fine features. However, the model must contain enough detail to identify the feature.

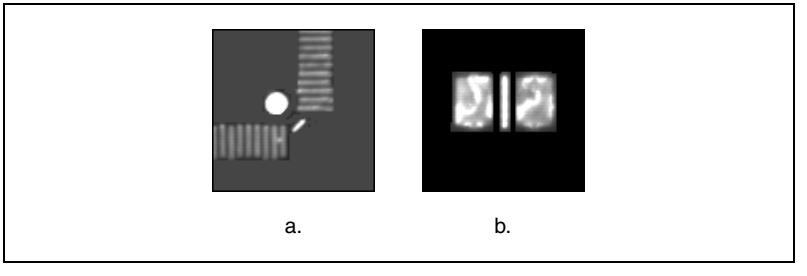


Figure 5-8. Feature Detail

Positional Information

A template with strong edges in both the x and y directions is easier to locate.

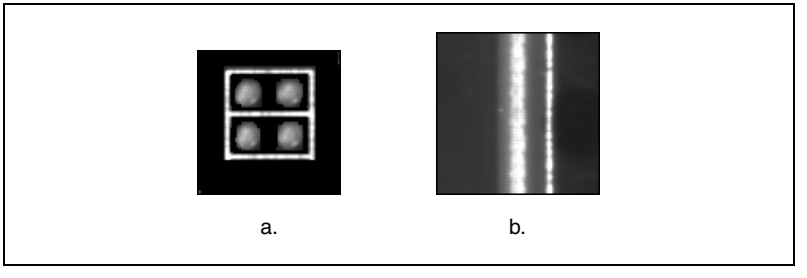


Figure 5-9. Positional Information

Background Information

Unique background information in a template improves search performance and accuracy.

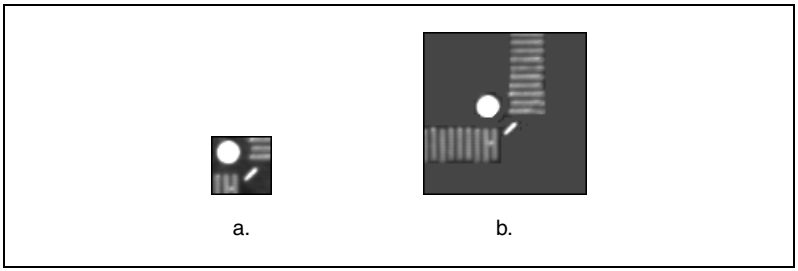


Figure 5-10. Background Information

Training the Pattern Matching Algorithm

After you create a good template image, the pattern matching algorithm has to learn the important features of the template. Use IMAQ Learn Pattern 2 to learn the template. The learning process depends on the type of matching that you expect to perform. If you do not expect the instance of the template in the image to rotate or change its size, the pattern matching algorithm has to learn only those features from the template that are necessary for shift-invariant matching. However, if you want to match the template at any orientation, the learning process must consider the possibility of arbitrary orientations. Use the IMAQ Setup Learn Pattern 2 VI (**Machine Vision»Searching and Matching**) to specify which type of learning mode to use.

The learning process is usually time intensive because the algorithm attempts to find unique features of the template that allow for fast, accurate matching. The learning mode you choose also affects the speed of the learning process. Learning the template for shift-invariant matching is faster than learning for rotation-invariant matching. You also can save time by training the pattern matching algorithm offline and then saving the template image with IMAQ Write Image and Vision Info VI (**Machine Vision»Searching and Matching**).

Defining a Search Area

Two equally important factors define the success of a pattern matching algorithm: accuracy and speed. You can define a search area to reduce ambiguity in the search process. For example, if your image has multiple instances of a pattern and only one of them is required for the inspection

task, the presence of additional instances of the pattern can produce incorrect results. To avoid this, reduce the search area so that only the required pattern lies within the search area.

The time required to locate a pattern in an image depends on both the template size and the search area. By reducing the search area, you can reduce the required search time. Increasing the template size can improve the search time, but doing so reduces match accuracy if the larger template includes an excess of background information.

In many inspection applications, you have general information about the location of the fiducial. Use this information to define a search area. For example, in a typical component placement application, each printed circuit board (PCB) being tested may not be placed in the same location with the same orientation. The location of the PCB9 in various images can move and rotate within a known range of values, as illustrated in Figure 5-11. Figure 5-11a shows the template used to locate the PCB in the image. Figure 5-11b shows an image containing a PCB with a fiducial you want to locate. Notice the search area around the fiducial. If you know before the matching process begins that the PCB can shift or rotate in the image within a fixed range—as shown in Figure 5-11c and Figure 5-11d, respectively—you can limit the search for the fiducial to a small region of the image.

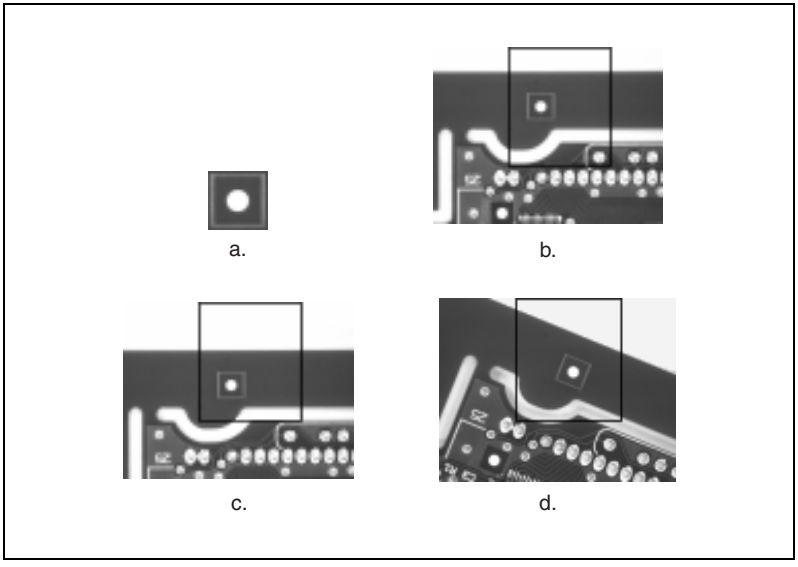


Figure 5-11. Selecting a Search Area for Grayscale Pattern Matching

Setting Matching Parameters and Tolerances

Every pattern matching algorithm makes assumptions about the images and pattern matching parameters used in machine vision applications. These assumptions work for a high percentage of the applications. However, there may be applications in which the assumptions used in the algorithm are not optimal. Knowing your particular application and the images you want to process is useful in selecting the pattern matching parameters. Use the IMAQ Setup Match Pattern 2 VI (**Machine Vision» Searching and Matching**) to set the following parameters that influence the NI Vision pattern matching algorithm: match mode, minimum contrast, and rotation angle ranges.

Match Mode

Set the match mode to control how the pattern matching algorithm treats the template at different orientations. If you expect the orientation of valid matches to vary less than ±5° from the template, set the **Match Mode** control to **Shift Invariant**. Otherwise, set **Match Mode** to **Rotation Invariant**. Shift-invariant matching is faster than rotation-invariant matching.

Minimum Contrast

The pattern matching algorithm ignores all image regions in which contrast values fall below a set minimum contrast value. Contrast is the difference between the smallest and largest pixel values in a region. Set the **Minimum Contrast** control to slightly below the contrast value of the search area with the lowest contrast.

You can set the minimum contrast to potentially increase the speed of the pattern matching algorithm. If the search image has high contrast overall but contains some low contrast regions, set a high minimum contrast value to exclude all areas of the image with low contrast. Excluding these areas significantly reduces the area in which the pattern matching algorithm must search. However, If the search image has low contrast throughout, set a low minimum contrast to ensure that the pattern matching algorithm looks for the template in all regions of the image.

Rotation Angle Ranges

If you know that the pattern rotation is restricted to a certain range—for example, between –15° and 15°—provide this restriction information to the pattern matching algorithm in the **Rotation Angle Ranges** control. This information improves your search time because the pattern matching algorithm looks for the pattern at fewer angles. Refer to Chapter 12,

Pattern Matching, of the *NI Vision Concepts Manual* for information about pattern matching.

Testing the Search Algorithm on Test Images

To determine if your selected template or reference pattern is appropriate for your machine vision application, test the template on a few test images by using the IMAQ Match Pattern 2 VI (**Machine Vision»Searching and Matching**). These test images should reflect the images generated by your machine vision application during true operating conditions. If the pattern matching algorithm locates the reference pattern in all cases, you have selected a good template. Otherwise, refine the current template, or select a better template until both training and testing are successful.

Using a Ranking Method to Verify Results

The manner in which you interpret the IMAQ Match Pattern 2 results depends on your application. For typical alignment applications, such as finding a fiducial on a wafer, the most important information is the position and location of the best match. Use the **Position** and **Bounding Box** elements of the **Matches** indicator to get the position and location of a match.

In inspection applications, such as optical character verification (OCV), the score of the best match is more useful. The score of a match returned by the pattern matching algorithm is an indicator of the closeness between the original pattern and the match found in the image. A high score indicates a very close match, while a low score indicates a poor match. The score can be used as a gauge to determine if a printed character is acceptable. Use the **Score** element of the **Matches** indicator to get the score corresponding to a match.

Finding Points Using Geometric Matching

The geometric matching algorithm in NI Vision locates regions in a grayscale image that match a model, or template, of a reference pattern. Geometric matching is specialized to locate templates that are characterized by distinct geometric or shape information. Geometric matching finds template matches regardless of lighting variation, blur, noise, occlusion, and geometric transformations such as shifting, rotation, or scaling of the template. Geometric matching returns the location of the center of the template, the template orientation, and the scale of the template.

Complete the following generalized steps to find features in an image using geometric matching.

1. Define a reference or fiducial pattern to use as a template image.
2. Use the reference pattern to train the geometric matching algorithm with the NI Vision Template Editor. Go to **Start»All Programs» National Instruments»Vision»Template Editor** to launch the NI Vision Template Editor.
3. Define an image or an area of an image as the search area. A small search area can reduce the time to find the features.

4. Set the tolerances and parameters to specify how the algorithm operates at run time using IMAQ Setup Match Geometric Pattern.

5. Test the search algorithm on test images using IMAQ Match Geometric Pattern.

Defining and Creating Effective Template Images

The selection of an effective template image plays a critical part in obtaining good results. Because the template image represents the pattern that you want to find, make sure that all the important and unique characteristics of the pattern are well defined in the image.

The geometric matching algorithm is optimized for locating objects with good geometric information. Figure 5-12 shows examples of template images with good geometric or shape information.

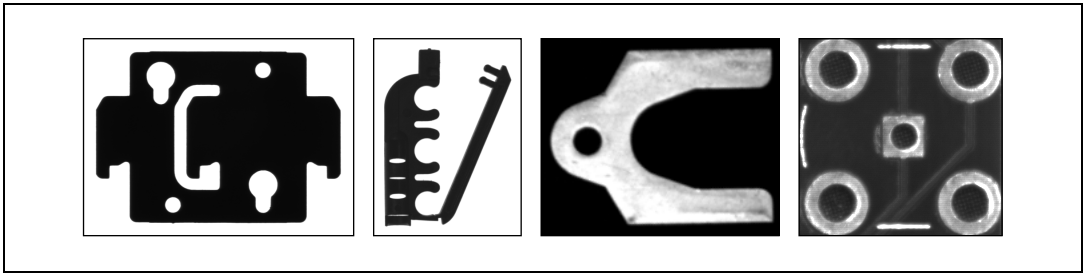
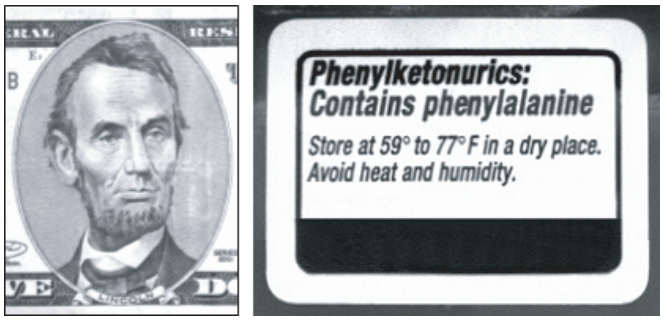


Figure 5-12. Examples of Objects on which Geometric Matching is Designed to Work

Geometric matching is not suited for template images that are predominantly defined by grayscale or texture information. Figure 5-13 shows examples of template images that do not have good geometric information. The template image in Figure 5-13a is characterized by the grayscale or texture information in the image. Figure 5-13b contains too many edges and will dramatically increase the time geometric matching takes to locate the template in an image.



a. b.

Figure 5-13. Examples of Objects for which Geometric Matching is Not Suited

Training the Geometric Matching Algorithm

After you create a good template image, the geometric matching algorithm has to learn the important features of the template. Use the NI Vision Template Editor to select the important features in the template. Geometric matching uses the curves found in the template image as the basis for the features that are used for matching. Use the NI Vision Template Editor to remove noisy or unimportant curves from the template image and keep only those curves that are important to locate the template in the image.

The template image of a pacemaker is shown in Figure 5-14a. Figure 5-14b shows all the curves that are found in the template image. These images are comprised of curves that are good for matching, such as those along the boundary of the pacemaker, as well as noisy or unimportant curves, such as those found around the barcode and areas of reflection in the image. If you learn the template with the unimportant curves, the geometric matching algorithm will treat these curves as important features that must be found within a valid match region. You can make the matching process more reliable by removing these curves before learning the template. Figure 5-14c shows the curves that are used to learn the template after the unimportant curves have been removed using the NI Vision Template Editor.

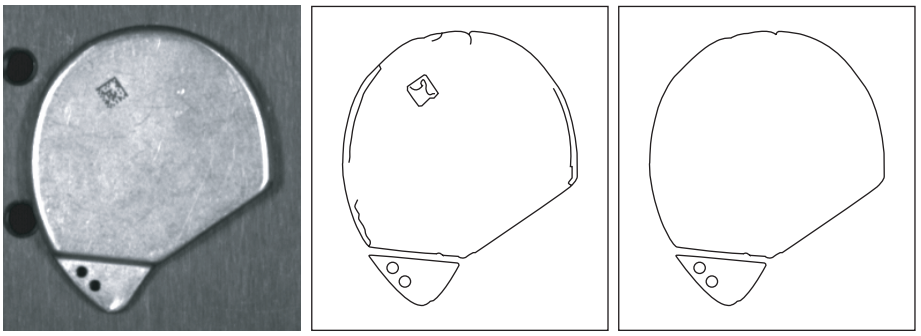


Figure 5-14. Selecting Good Features for Training a Template

Once the template image is trained, you can save the template to a file from within the NI Vision Template Editor. Use the IMAQ Read Image and Vision Info VI (**Machine Vision»Searching and Matching**) to read the template file within your application.

Refer to the *NI Vision Template Editor Help* for more information about training the Geometric Matching algorithm.

Setting Matching Parameters and Tolerances

The geometric matching algorithm makes assumptions about the images and geometric matching parameters used in machine vision applications. These assumptions work for a high percentage of the applications. However, there may be applications in which the assumptions used in the algorithm are not optimal. Knowing your particular application and the images you want to process is useful in selecting the pattern matching parameters. Use the IMAQ Setup Match Geometric Pattern VI (**Machine Vision»Searching and Matching**) to set the following parameters that influence the NI Vision geometric matching algorithm: match mode, rotation angle ranges, scale ranges, and occlusion ranges.

Match Mode

Set the match mode to control the conditions under which the geometric matching algorithm finds the template matches. If you expect the orientation of the valid matches to vary more than ±5° from the template, enable the **Rotation** option within **Match Mode**. If you expect the size of the valid matches to change more than ±5%, enable the **Scale** option within **Match Mode**. If you expect the valid matches to be partially covered or

missing, enable the **Occlusion** option in **Match Mode**. Disabling any of these options or limiting their ranges decreases the search time.

Rotation Angle Ranges

If you know the pattern is restricted to a certain range—for example, between –15° and 15°—provide this restriction information to the geometric matching algorithm by setting the **Rotation Angle (degrees)** option in the **Range Settings** control. This information improves your search time because the geometric matching algorithm looks for the pattern at fewer angles.

Scale Factor Ranges

When the **Scale** option in **Match Mode** is enabled, geometric matching searches for occurrences of the template in the image regardless of whether valid matches are of a different size. The default scale range is 75% to 125%. If you know that the scale range of the valid matches is restricted to a certain range—for example, between 90% and 110%—provide this restriction information to the geometric matching algorithm by setting the **Scale Factor (%)** option within the **Range Settings** control.

Occlusion Ranges

When the **Occlusion** option in **Match Mode** is enabled, geometric matching searches for occurrences of the template in the image, allowing for a specified percentage of the template to be occluded. The default occlusion range is 0% to 25%. If you know that the occlusion range of the valid matches is restricted to a certain range—for example, between 0% and 10%—provide this restriction information to the geometric matching algorithm by setting the **Occlusion (%)** option within the **Range Settings** control.

Refer to the Chapter 13, *Geometric Matching,* of the *NI Vision Concepts Manual* for more information about geometric matching.

Testing the Search Algorithm on Test Images

To determine if your selected template or reference pattern is appropriate for your machine vision application, test the template on a few test images by using the IMAQ Match Geometric Pattern VI (**Machine Vision» Searching and Matching**). These test images should reflect the images generated by your machine vision application during true operating conditions. If the geometric matching algorithm locates the reference pattern in all cases, you have selected a good template. Otherwise, refine

the current template, or select a better template until both training and testing are successful.

Using Multiple Template Images

Complete the following steps if your application requires locating multiple template images in a target image.

1. Use the methods described in *Defining and Creating Effective Template Images* and *Training the Geometric Matching Algorithm* to create and train each template image.

2. Use the IMAQ Learn Multiple Geometric Patterns VI **(Machine Vision»Searching and Matching)** to combine the templates you trained in the previous step into a **Multiple Geometric Template**.

If you want to perform the learning process off-line, use the IMAQ Write Multiple Geometric Template VI **(Machine Vision»Searching and Matching)** to save to file the multiple geometric templates created by this step. Use the IMAQ Read Multiple Geometric Template VI **(Machine Vision»Searching and Matching)** to read the multiple geometric template file within your application during the matching process.

3. Use the methods described in *Setting Matching Parameters and Tolerances* to define how each template matches the target image.

Note Even though the parameters that you create in this step are used only during the match phase, you can pass these parameters to IMAQ Learn Multiple Geometric Patterns VI during the learn phase. Doing this stores these parameters in the multiple geometric template, and they will be saved to file when you save the multiple geometric template. When you read the multiple geometric template file within your application, it will be part of the multiple geometric template, and will be used during the matching process. If you wish to override some of these parameters during the matching process, you can do so by passing the overriding parameters to the IMAQ Match Multiple Geometric Patterns VI.

4. Test the multiple template search algorithm on test images using the IMAQ Match Multiple Geometric Patterns VI **(Machine Vision» Searching and Matching)** to match all of the templates. The importance of this step is for the same reason explained in *Testing the Search Algorithm on Test Images*.

Finding Points Using Color Pattern Matching

Color pattern matching algorithms provide a quick way to locate objects when color is present. Use color pattern matching if your images have the following qualities:

- The object you want to locate has color information that is very different from the background, and you want to find a very precise location of the object in the image.
- The object to locate has grayscale properties that are very difficult to characterize or that are very similar to other objects in the search image. In such cases, grayscale pattern matching can give inaccurate results. If the object has color information that differentiates it from the other objects in the scene, color provides the machine vision software with the additional information to locate the object.

Color pattern matching returns the location of the center of the template and the template orientation. Complete the following general steps to find features in an image using color pattern matching:

1. Define a reference or fiducial pattern in the form of a template image.
2. Use the reference pattern to train the color pattern matching algorithm with IMAQ Setup Learn Color Pattern.
3. Define an image or an area of an image as the search area. A small search area can reduce the time to find the features.
4. Set the **Feature Mode** control to **Color and Shape**.
5. Set the tolerances and parameters to specify how the algorithm operates at run time using IMAQ Setup Match Color Pattern.
6. Test the search tool on test images using IMAQ Match Color Pattern.
7. Verify the results using a ranking method.

Defining and Creating Good Color Template Images

The selection of a good template image plays a critical part in obtaining accurate results with the color pattern matching algorithm. Because the template image represents the color and the pattern that you want to find, make sure that all the important and unique characteristics of the pattern are well defined in the image.

Several factors are critical in creating a template image. These critical factors include color information, symmetry, feature detail, positional information, and background information. Refer to the *Defining and Creating Effective Template Images* section of this chapter for more information about some of these factors.

Color Information

A template with colors that are unique to the pattern provides better results than a template that contains many colors, especially colors found in the background or other objects in the image.

Symmetry

A rotationally symmetric template in the luminance plane is less sensitive to changes in rotation than one that is rotationally asymmetric.

Feature Detail

A template with relatively coarse features is less sensitive to variations in size and rotation than a model with fine features. However, the model must contain enough detail to identify it.

Positional Information

A color template whose luminance plane contains strong edges in both the x and y directions is easier to locate.

Background Information

Unique background information in a template improves search performance and accuracy during the grayscale pattern matching phase. This requirement could conflict with the color information requirement of color pattern matching because background colors may interfere with the color location phase. Avoid this problem by choosing a template with sufficient background information for grayscale pattern matching while specifying the exclusion of the background color during the color location phase. Refer to the *Training the Color Pattern Matching Algorithm* section of this chapter for more information about how to ignore colors.

Training the Color Pattern Matching Algorithm

After you have created a good template image, the color pattern matching algorithm needs to learn the important features of the template.
The learning process depends on the type of matching that you expect to perform. By default, the color pattern matching algorithm learns only those features from the template that are necessary for shift-invariant matching. However, if you want to match the template at any orientation, the learning process must consider the possibility of arbitrary orientations.

Use the IMAQ Setup Learn Color Pattern VI **(Machine Vision»Searching and Matching)** to specify which type of learning mode to use. Then use the IMAQ Learn Color Pattern VI **(Machine Vision»Searching and Matching)** to learn the template.

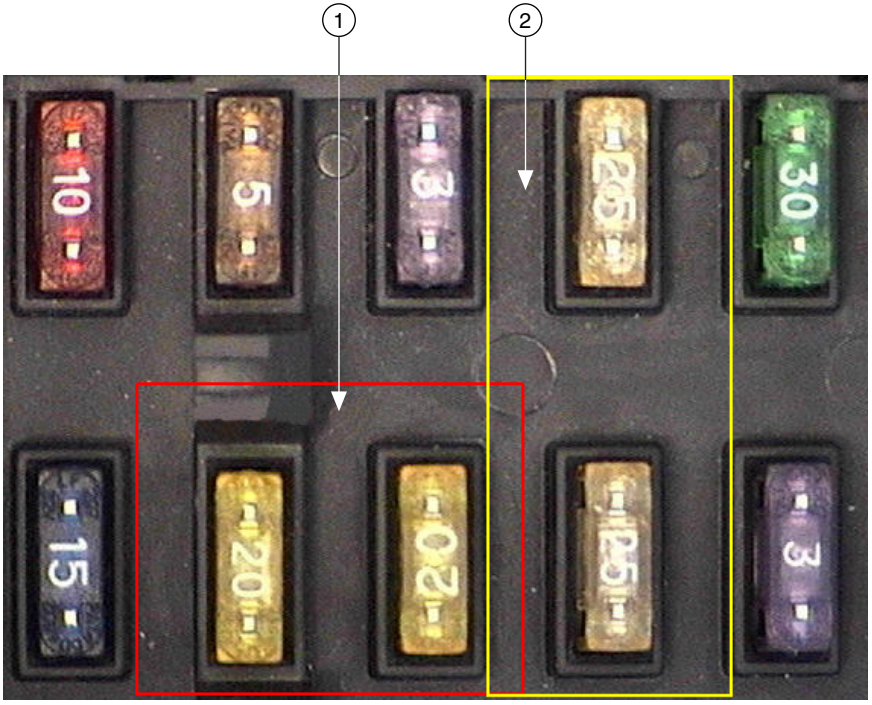
Exclude colors in the template that you are not interested in using during the search phase. Ignore colors that make your template difficult to locate. When a template differs from several regions of the search image by only its primary color or colors, consider ignoring the predominant common color to improve search performance. Typically, the predominant color is the background color of the template.

Use the IMAQ Setup Learn Color Pattern VI to ignore colors. You can ignore certain predefined colors by using **Ignore Black and White**. To ignore other colors, first learn the colors to ignore using IMAQ ColorLearn. Then set the **Ignore Color Spectra** control of the IMAQ Setup Learn Color Pattern VI to the resulting color spectrum.

The learning process is time-intensive because the algorithm attempts to find unique features of the template that allow for fast, accurate matching. However, you can train the pattern matching algorithm offline, and save the template image using the IMAQ Write Image and Vision Info VI (**Machine Vision»Searching and Matching**).

Defining a Search Area

Two equally important factors define the success of a color pattern matching algorithm: accuracy and speed. You can define a search area to reduce ambiguity in the search process. For example, if your image has multiple instances of a pattern and only one instance is required for the inspection task, the presence of additional instances of the pattern can produce incorrect results. To avoid this, reduce the search area so that only the required pattern lies within the search area. For instance, in the fuse box inspection example, use the location of the fuses to be inspected to define the search area. Because the inspected fuse box may not be in the exact location or have the same orientation in the image as the previous one, the search area you define should be large enough to accommodate these variations in the position of the box. Figure 5-15 shows how search areas can be selected for different objects.



1	Search Area for 20 Amp Fuses	2	Search Area for 25 Amp Fuses

Figure 5-15. Selecting a Search Area for Color Pattern Matching

The time required to locate a pattern in an image depends on both the template size and the search area. By reducing the search area, you can reduce the required search time. Increasing the template size can improve the search time, but doing so reduces match accuracy if the larger template includes an excess of background information.

Setting Matching Parameters and Tolerances

Every color pattern matching algorithm makes assumptions about the images and color pattern matching parameters used in machine vision applications. These assumptions work for a high percentage of the applications. However, there may be applications in which the assumptions used in the algorithm are not optimal. In such cases you must modify the color pattern matching parameters. Knowing your particular application and the images you want to process is useful in selecting the pattern matching parameters. Use the IMAQ Setup Match Color Pattern VI

(**Machine Vision»Searching and Matching**) to set the following parameters that influence color pattern matching: color sensitivity, search strategy, color score weight, ignore colors, minimum contrast, and rotation angle ranges.

Color Sensitivity

Use the color sensitivity to control the granularity of the color information in the template image. If the background and objects in the image contain colors that are very close to colors in the template image, use a higher color sensitivity setting. A higher sensitivity setting distinguishes colors with very close hue values. Three color sensitivity settings are available in NI Vision: low, medium, and high. Use the default low setting if the colors in the template are very different from the colors in the background or other objects that you are not interested in. Increase the color sensitivity settings as the color differences decrease. Use the **Color Sensitivity** control of the IMAQ Setup Match Color Pattern VI to set the color sensitivity. Refer to Chapter 15, *Color Inspection*, of the *NI Vision Concepts Manual* for more information about color sensitivity.

Search Strategy

Use the search strategy to optimize the speed of the color pattern matching algorithm. The search strategy controls the step size, subsampling factor, and percentage of color information used from the template.

Choose from these strategies:

- Conservative—Uses a very small step size, the least amount of subsampling, and all the color information present in the template. The conservative strategy is the most reliable method to look for a template in any image at potentially reduced speed.

Note Use the conservative strategy if you have multiple targets located very close to each other in the image.

- Balanced—Uses values in between the aggressive and conservative strategies.
- Aggressive—Uses a large step size, a lot of subsampling, and all the color spectrum information from the template.
- Very Aggressive—Uses the largest step size, the most subsampling, and only the dominant color from the template to search for the template. Use this strategy when the color in the template is almost uniform, the template is well contrasted from the background and there

is a good amount of separation between different occurrences of the template in the image. This strategy is the fastest way to find templates in an image.

Decide on the best strategy by experimenting with the different options. Use the **Search Strategy** control to select a search strategy.

Color Score Weight

When you search for a template using both color and shape information, the color and shape scores generated during the match process are combined to generate the final color pattern matching score. The color score weight determines the contribution of the color score to the final color pattern matching score. If the color information of the templates is superior to the shape information of the template, set the weight higher. For example, if you use a weight of 1000, the algorithm finds each match by using both color and shape information and then ranks the matches based entirely on their color scores. If the weight is 0, the matches are still found using color and shape information, but they are ranked based entirely on their shape scores. Use the **Color Score Weight** control to set the color score weight.

Minimum Contrast

Use the minimum contrast to increase the speed of the color pattern matching algorithm. The color pattern matching algorithm ignores all image regions where grayscale contrast values fall beneath a minimum contrast value. Use the **Minimum Contrast** control to set the minimum contrast. Refer to the *Setting Matching Parameters and Tolerances* section of this chapter for more information about minimum contrast.

Rotation Angle Ranges

Refer to the *Setting Matching Parameters and Tolerances* section of this chapter for information about rotation angle ranges.

Testing the Search Algorithm on Test Images

Refer to the previous *Testing the Search Algorithm on Test Images* section of this chapter for information about testing the search algorithm.

Finding Points Using Color Location

Color location algorithms provide a quick way to locate regions in an image with specific colors. Use color location when your application has the following characteristics:

- Requires the location and the number of regions in an image with their specific color information
- Relies on the cumulative color information in the region instead of the color arrangement in the region
- Does not require the orientation of the region
- Does not always require the location with subpixel accuracy
- Does not require shape information for the region

Complete the following general steps to find features in an image using color location:

1. Define a reference pattern in the form of a template image.
2. Use the reference pattern to train the color location algorithm with IMAQ Learn Color Pattern.

3. Define an image or an area of an image as the search area. A small search area can reduce the time to find the features.

4. Set the **Feature Mode** control of IMAQ Setup Learn Color Pattern to **Color**.
5. Set the tolerances and parameters to specify how the algorithm operates at run time using IMAQ Setup Match Color Pattern.

6. Test the color location algorithm on test images using IMAQ Match Color Pattern.
7. Verify the results using a ranking method.

You can save the template image using the IMAQ Write Image and Vision Info VI (**Machine Vision»Searching and Matching**).

Convert Pixel Coordinates to Real-World Coordinates

The measurement points you located with edge detection and pattern matching are in pixel coordinates. If you need to make measurements using real-world units, use the IMAQ Convert Pixel to Real World VI (**Vision Utilities»Calibration**) to convert the pixel coordinates into real-world units.

Make Measurements

You can make different types of measurements either directly from the image or from points that you detect in the image.

Making Distance Measurements

Use clamp VIs (**Machine Vision»Measure Distances**) to measure the separation between two edges in a rectangular search region. Specify the parameters for edge detection and the separation between the search lines that you want to use within the search region to find the edges.

Note Use the IMAQ Select Rectangle VI (**Machine Vision»Select Region of Interest**) to generate a valid input search region for the clamp VIs.

First the VIs use the rake function to detect points along two edges of the object under inspection. Then the VIs compute the distance between the points detected on the edges along each search line of the rake. The VIs return the largest or smallest distance in either the horizontal or vertical direction, and they output the coordinates of all the edge points that they find.

The following list describes the available clamp VIs:

- IMAQ Clamp Horizontal Max—Measures the largest horizontal separation between two edges in a rectangular search region.
- IMAQ Clamp Horizontal Min—Finds the smallest horizontal separation between two vertically oriented edges.
- IMAQ Clamp Vertical Max—Finds the largest vertical separation between two horizontally oriented edges.
- IMAQ Clamp Vertical Min—Finds the smallest vertical separation between two horizontally oriented edges.

Use the IMAQ Point Distances VI (**Machine Vision»Analytic Geometry**) to compute the distances between consecutive pairs of points in an array of points. You can obtain these points from the image using any one of the feature detection methods described in the *Find Measurement Points* section of this chapter.

Making Analytic Geometry Measurements

Use the following VIs (**Machine Vision»Analytic Geometry**) to make geometrical measurements from the points you detect in the image:

- IMAQ Fit Line—Fits a line to a set of points and computes the equation of the line.

- IMAQ Fit Circle 2—Fits a circle to a set of at least three points and computes its area, perimeter and radius.

- IMAQ Fit Ellipse 2—Fits an ellipse to a set of at least six points and computes its area, perimeter, and the lengths of its major and minor axis.

- IMAQ Lines Intersection—Finds the angle and point of intersection of two lines specified by their start and end points.

- IMAQ Perpendicular Line—Finds the perpendicular line and distance from a point to a line.

- IMAQ Bisecting Line—Finds the line that bisects the angle formed by two lines.

- IMAQ Mid Line—Finds the line that is midway between a point and a line and is parallel to the line.

- IMAQ Polygon Area—Calculates the area of a polygon specified by its vertex points.

Making Instrument Reader Measurements

Use the following VIs to make measurements based on the values obtained by meter and LCD readers:

- Use the IMAQ Get Meter or IMAQ Get Meter 2 VIs (**Machine Vision»Instrument Readers**) to calibrate a meter or gauge that you want to read.

  – IMAQ Get Meter calibrates the meter using the initial position and full-scale position of the needle. This VI calculates the position of the base of the needle and the arc traced by the tip of the needle.

  – IMAQ Get Meter 2 calibrates the meter using three points on the meter: the base of the needle, the tip of the needle at its initial position, and the tip of the needle at its full-scale position. This VI calculates the position of the points along the arc covered by the tip of the needle.

- Use the IMAQ Read Meter VI (**Machine Vision»Instrument Readers**) to read the position of the needle using the base of the needle and the array of points on the arc traced by the tip of the needle.

- Use the IMAQ Get LCD ROI VI (**Machine Vision»Instrument Readers**) to calculate the ROI around each digit in an LCD or LED. To find the area of each digit, all the segments of the indicator must be activated.

- Use the IMAQ Read Single Digit VI (**Machine Vision»Instrument Readers**) to read one digit of an LCD or LED. Use the IMAQ Read LCD VI (**Machine Vision»Instrument Readers**) to read multiple digits of an LCD or LED.

Identify Parts Under Inspection

In addition to making measurements after you set regions of inspection, you also can identify parts using classification, optical character recognition (OCR), and barcode reading.

Classifying Samples

Use classification to identify an unknown object by comparing a set of its significant features to a set of features that conceptually represent classes of known objects. Typical applications involving classification include the following:

- Sorting—Sorts objects of varied shapes. For example, sorting different mechanical parts on a conveyor belt into different bins.
- Inspection—Inspects objects by assigning each object an identification score and then rejecting objects that do not closely match members of the training set.

Before you classify objects, you must train the Classifier Session with samples of the objects using the NI Classification Training Interface. Go to **Start»All Programs»National Instruments»Vision»Classification Training** to launch the NI Classification Training Interface.

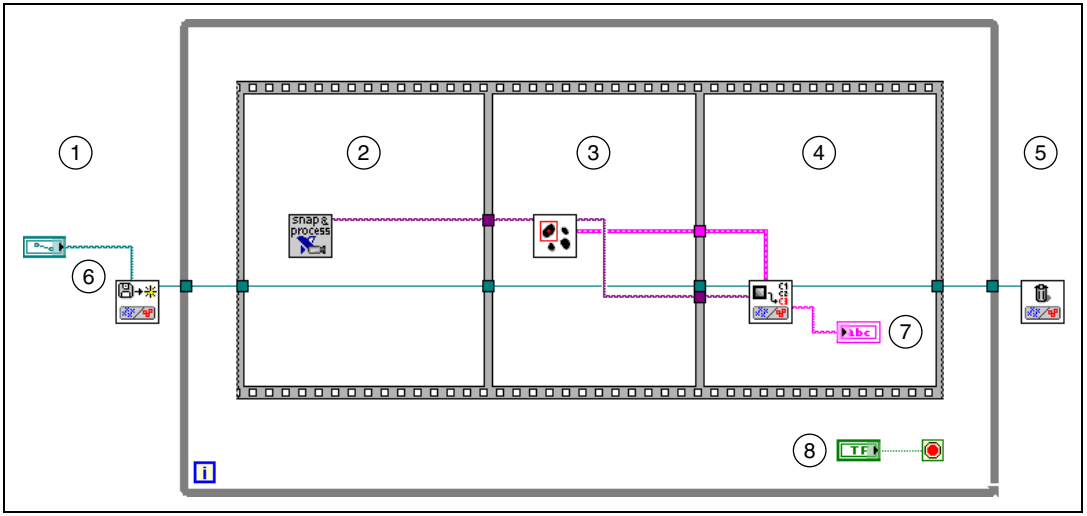
After you have trained samples of the objects you want to classify, use the following VIs to classify the objects:

1. In the initialization part the of code, use IMAQ Read Classifier File (**Machine Vision»Classification**) to read in a Classifier that you created using the NI Classification Training Interface.

2. Use IMAQ Classify (**Machine Vision»Classification**) to classify the object inside the ROI of the image under inspection into one of the classes you created using the NI Classification Training Interface.

3. Use IMAQ Dispose Classifier (**Machine Vision»Classification**) to free the resources that the Classifier Session used.

Figure 5-16 shows LabVIEW pseudocode of a typical classification VI.



1	Read a Classifier File	5	Dispose of the Classifier Session
2	Acquire and Preprocess the Image	6	Path to the Trained Classifier File
3	Locate the Sample to Classify	7	Class
4	Classify the Sample	8	Stop

Figure 5-16. Sample Code of a Classification Application

Reading Characters

Use OCR to read text and/or characters in an image. Typical uses for OCR in an inspection application include identifying or classifying components.

Before you read text and/or characters in an image, you must train the OCR Session with samples of the characters using the NI OCR Training Interface. Go to **Start»All Programs»National Instruments»Vision»OCR Training** to launch the NI OCR Training Interface.

After you have trained samples of the characters you want to read, use the following VIs to read the characters:

1. In the initialization part the of code, use IMAQ OCR Read Character Set File (**Machine Vision»OCR**) to read in a session that you created using the NI OCR Training Interface.

2. Use IMAQ OCR Read Text (**Machine Vision»OCR**) to read the characters inside the ROI of the image under inspection.

3. Use IMAQ OCR Dispose Session (**Machine Vision»OCR**) to free the resources that the OCR Session used.

Reading Barcodes

Use barcode reading VIs to read values encoded into 1D barcodes, Data Matrix barcodes, and PDF417 barcodes.

Reading 1D Barcodes

Use IMAQ Read Barcode (**Machine Vision»Instrument Readers**) to read values encoded in a 1D barcode. Locate the barcode in the image using one of the techniques described in the *Locate Objects to Inspect* and *Set Search Areas* section of this chapter. Then pass the ROI Descriptor of the location into IMAQ Read Barcode. Specify the type of 1D barcode in the application using the **Barcode Type** control.

NI Vision supports the following 1D barcode types: Codabar, Code 39, Code 93, Code 128, EAN 8, EAN 13, Interleaved 2 of 5, MSI, and UPCA.

Reading Data Matrix Barcodes

Use IMAQ Read Data Matrix Barcode 2 (**Machine Vision»Instrument Readers**) to read values encoded in a Data Matrix barcode. The VI can automatically determine the appropriate search options for your application. However, you can improve the performance of the application by specifying control values specific to your application.

IMAQ Read Data Matrix Barcode 2 can locate automatically the Data Matrix barcode in an image. However, you can improve the inspection performance by locating the barcode using one of the techniques described in Chapter 5, *Performing Machine Vision Tasks*, and then passing in ROI Descriptors of the locations into IMAQ Read Data Matrix Barcode 2.

Reading PDF417 Barcodes

Use IMAQ Read PDF417 Barcode (**Machine Vision»Instrument Readers**) to read values encoded in a PDF417 barcode.

IMAQ Read PDF417 Barcode can automatically locate one or multiple PDF417 barcodes in an image. However, you can improve the inspection performance by locating the barcodes using one of the techniques described in Chapter 5, *Performing Machine Vision Tasks*, and then passing in ROI Descriptors of the locations into IMAQ Read PDF417 Barcode.

Tip If you need to read only one PDF417 barcode per image, set **Search Mode** to **Single Barcode, Conservative** to increase the speed of the VI.

Inspect Image for Defects

You can inspect an image for defects by comparing it to a golden template and by verifying characters in the image.

Compare to Golden Template

Use IMAQ Compare Golden Template (**Vision»Machine Vision»Inspection**) to inspect your image based on differences in intensity. Use the NI Vision Template Editor to learn golden templates for inspection. Go to **Start»All Programs»National Instructions» Vision»Template Editor** to launch the NI Vision Template Editor. Use the methods described in Chapter 4, *Performing Particle Analysis*, to analyze the resulting binary image.

Verify Characters

Use character verification functions to verify characters in your image.

Before you verify text, you must train the OCR Session with samples of the characters using the NI OCR Training Interface. Go to **Start»All Programs»National Instruments»Vision»OCR Training** to launch the NI OCR Training Interface. You must then designate reference characters for each of the character classes in your character set. The characters you want to verify are compared against these reference characters and are assigned a score based on this comparison.

After you have trained the samples of the characters and assigned the reference character, use the following functions to verify the characters.

1. In the initialization part the of code, use IMAQ OCR Read Character Set File (**Machine Vision»OCR**) to read in a session that you created using the NI OCR Training Interface.

2. Use IMAQ OCR Verify Text (**Machine Vision»OCR**) to verify the characters inside the ROI(s) of the image under inspection.

3. Use IMAQ OCR Dispose Session (**Machine Vision»OCR**) to free the resources that the OCR Session used.

Display Results

You can overlay the results obtained at various stages of your inspection process on the inspection image. NI Vision attaches the information that you want to overlay to the image, but it does not modify the image. The overlay appears every time you display the image.

Use the following VIs (**Vision Utilities»Overlay**) to overlay search regions, inspection results, and other information, such as text and bitmaps.

- IMAQ Overlay Points—Overlays points on an image. Specify a point by its x-coordinate and y-coordinate.

- IMAQ Overlay Line—Overlays a line on an image. Specify a line by its start and end points.

- IMAQ Overlay Multiple Lines—Overlays multiple lines on an image.

- IMAQ Overlay Rectangle—Overlays a rectangle on an image.

- IMAQ Overlay Oval—Overlays an oval or a circle on the image.

- IMAQ Overlay Arc—Overlays an arc on the image.

- IMAQ Overlay Bitmap—Overlays a bitmap on the image.

- IMAQ Overlay Text—Overlays text on an image.

- IMAQ Overlay ROI—Overlays an ROI described by the ROI Descriptor on an image.

To use these VIs, pass in the image on which you want to overlay information and the information that you want to overlay.

Tip You can select the color of the overlays using the previous VIs.

You can configure the following processing VIs to overlay different types of information about the inspection image:

- IMAQ Find Vertical Edge (**Machine Vision»Locate Edges**)

- IMAQ Find Horizontal Edge (**Machine Vision»Locate Edges**)

- IMAQ Find Circular Edge (**Machine Vision»Locate Edges**)

- IMAQ Find Concentric Edge (**Machine Vision»Locate Edges**)

- IMAQ Clamp Horizontal Max (**Machine Vision»Measure Distances**)

- IMAQ Clamp Horizontal Min (**Machine Vision»Measure Distances**)

- IMAQ Clamp Vertical Max (**Machine Vision»Measure Distances**)

- IMAQ Clamp Vertical Min (**Machine Vision»Measure Distances**)

- IMAQ Find Pattern (**Machine Vision»Find Patterns**)

- IMAQ Count Object (**Machine Vision»Count and Measure Objects**)

- IMAQ Find CoordSys (Rect) (**Machine Vision»Coordinate Systems**)

- IMAQ Find CoordSys (2 Rects) (**Machine Vision»Coordinate Systems**)

- IMAQ Find CoordSys (Pattern) (**Machine Vision»Coordinate Systems**)

The following list contains the kinds of information you can overlay on the above VIs:

- Search area input into the VI
- Search lines used for edge detection
- Edges detected along the search lines
- Bounding rectangle of particles
- Center of particles
- Result of the VI

Select the information you want to overlay by enabling the corresponding Boolean control of the VI.

Use the IMAQ Clear Overlay VI (**Vision Utilities»Overlay**) to clear any previous overlay information from the image. Use the IMAQ Write Image and Vision Info VI (**Vision Utilities»Overlay**) to save an image with its overlay information to a file. You can read the information from the

file into an image using the IMAQ Read Image and Vision Info VI (**Vision Utilities»Overlay**).

Note As with calibration information, overlay information is removed from an image when the image size or orientation changes.

6

Calibrating Images

This chapter describes how to calibrate your imaging system, save calibration information, and attach calibration information to an image.

After you set up your imaging system, you may want to calibrate your system. If your imaging setup is such that the camera axis is perpendicular or nearly perpendicular to the object under inspection, and your lens has no distortion, use simple calibration. With simple calibration, you do not need to learn a template. Instead, you define the distance between pixels in the horizontal and vertical directions using real-world units.

If your camera axis is not perpendicular to the object under inspection, use perspective calibration to calibrate your system. If your lens is distorted, use nonlinear distortion calibration.

Perspective and Nonlinear Distortion Calibration

Perspective errors and lens aberrations cause images to appear distorted. This distortion misplaces information in an image, but it does not necessarily destroy the information in the image. Calibrate your imaging system if you need to compensate for perspective errors or nonlinear lens distortion.

Complete the following general steps to calibrate your imaging system:

1. Define a calibration template.
2. Define a reference coordinate system.
3. Learn the calibration information.

After you calibrate your imaging system, you can attach the calibration information to an image. Refer to the *Attach Calibration Information* section of this chapter for more information. Then, depending on your needs, you can perform one of the following steps:

- Use the real-world measurements options on the Particle Analysis and Particle Analysis Reports VIs to get real-world particle shape parameters without correcting the image.

- Use the calibration information to convert pixel coordinates to real-world coordinates without correcting the image.
- Create a distortion-free image by correcting the image for perspective errors and lens aberrations.

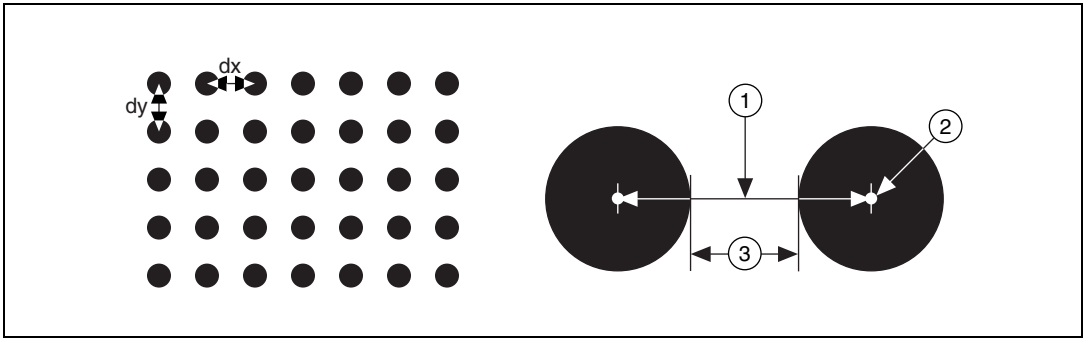
Refer to Chapter 6, *Calibrating Images*, for more information about applying calibration information before making measurements.

Defining a Calibration Template

You can define a calibration template by supplying an image of a grid or providing a list of pixel coordinates and their corresponding real-world coordinates. This section discusses, in detail, the grid method of defining a calibration template in detail.

A calibration template is a user-defined grid of circular dots. As shown in Figure 6-1, the grid has constant spacings in the x and y directions. You can use any grid, but follow these guidelines for best results:

- The displacement in the x and y directions should be equal ($dx = dy$).
- The dots should cover the entire required working area.
- The radius of the dots in the acquired image should be six to 10 pixels.
- The center-to-center distance between dots in the acquired image should range from 18 to 32 pixels, as shown in Figure 6-1.
- The minimum distance between the edges of the dots in the acquired image should be six pixels, as shown in Figure 6-1.



1 Center-to-Center Distance	2 Center of Grid Dots	3 Distance Between Dot Edges

Figure 6-1. Defining a Calibration Grid

Note Click **Start»All Programs»National Instruments»Vision»Documentation» Calibration Grid** to use the calibration grid installed with NI Vision. The dots have radii of 2 mm and center-to-center distances of 1 cm. Depending on your printer, these measurements may change by a fraction of a millimeter. You can purchase highly accurate calibration grids from optics suppliers, such as Edmund Industrial Optics.

Defining a Reference Coordinate System

To express measurements in real-world units, you need to define a coordinate system in the image of the grid. Use the **Calibration Axis Info** control of the IMAQ Set Simple Calibration VI (**Vision Utilities» Calibration**) to define a coordinate system by its origin, angle, and axis direction.

The origin, expressed in pixels, defines the center of your coordinate system. The angle specifies the orientation of your coordinate system with respect to the angle of the topmost row of dots in the grid image. The calibration procedure automatically determines the direction of the horizontal axis in the real world. The vertical axis direction can either be indirect, as shown in Figure 6-2a, or direct, as shown in Figure 6-2b.

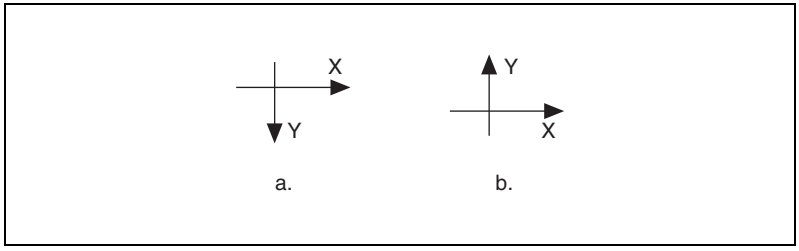
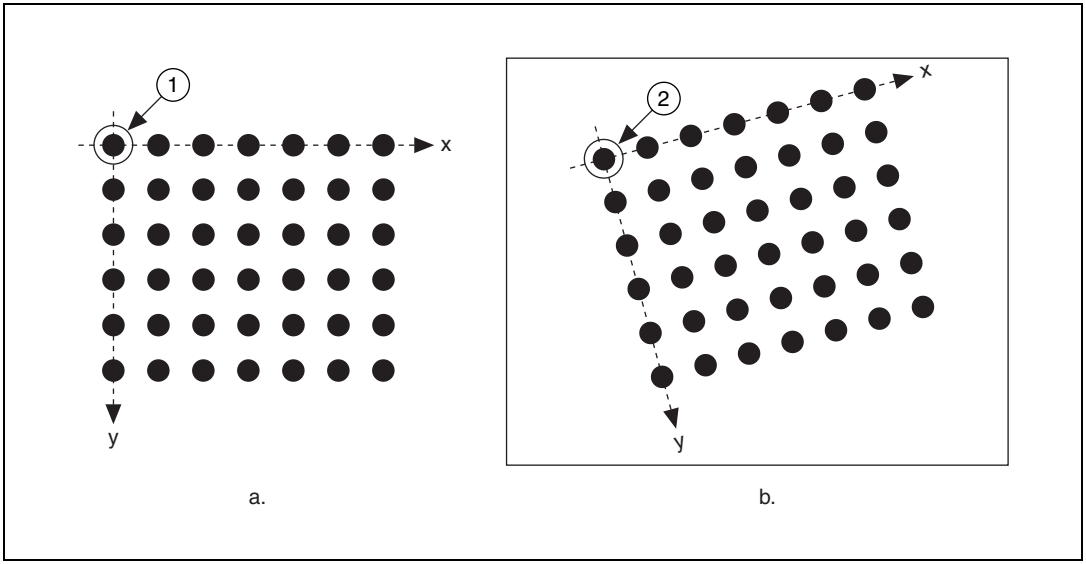


Figure 6-2. Axis Direction in the Image Plane

If you do not specify a coordinate system, the calibration process defines a default coordinate system. If you specify a grid for the calibration process, the software defines the following default coordinate system, as shown in Figure 6-3:

- The origin is placed at the center of the leftmost, topmost dot in the calibration grid.
- The angle is set to 0°. This aligns the x-axis with the first row of dots in the grid, as shown in Figure 6-3b.
- The axis direction is set to **Indirect**. This aligns the y-axis to the first column of the dots in the grid, as shown in Figure 6-3b.



1 Origin of a Calibration Grid in the Real World 2 Origin of the Same Calibration Grid in an Image

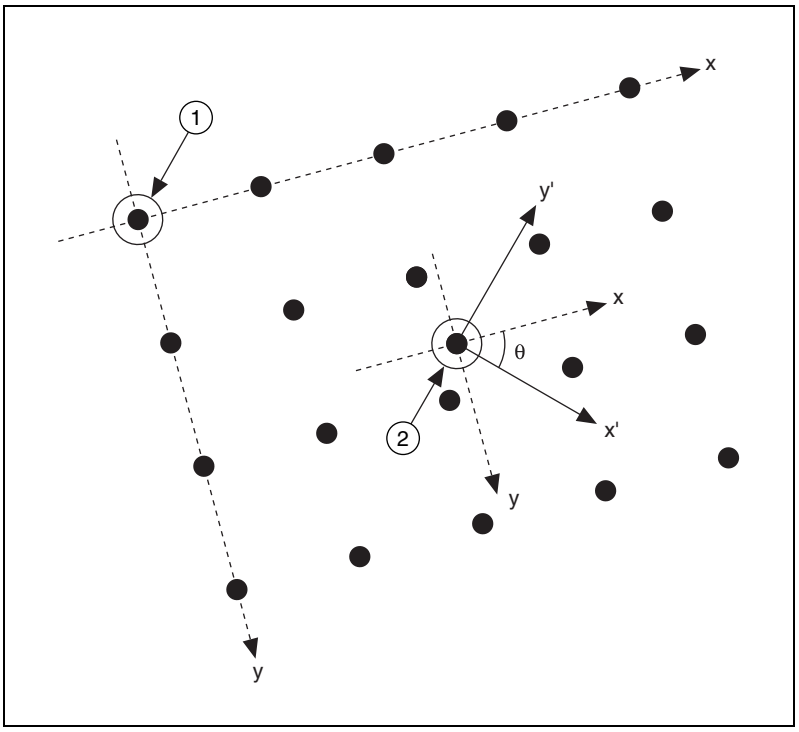
Figure 6-3. A Calibration Grid and an Image of the Grid

If you specify a list of points instead of a grid for the calibration process, the software defines a default coordinate system, as follows:

1. The origin is placed at the point in the list with the lowest x-coordinate value and then the lowest y-coordinate value.
2. The angle is set to 0°.
3. The axis direction is set to **Indirect**.

If you define a coordinate system yourself, carefully consider the needs of your application.

- Express the origin in pixels. Always choose an origin location that lies within the calibration grid so that you can convert the location to real-world units.
- Specify the angle as the angle between the x-axis of the new coordinate system (x') and the top row of dots (x), as shown in Figure 6-4. If your imaging system exhibits nonlinear distortion, you cannot visualize the angle as you can in Figure 6-4 because the dots do not appear in straight lines.



1 Default Origin in a Calibration Grid Image 2 User-Defined Origin

Figure 6-4. Defining a Coordinate System

Learning Calibration Information

After you define a calibration grid and reference axis, acquire an image of the grid using the current imaging setup. Refer to the *Acquire or Read an Image* section of Chapter 2, *Getting Measurement-Ready Images*, for information about acquiring images. The grid does not need to occupy the entire image. You can choose a region within the image that contains the grid. After you acquire an image of the grid, learn the calibration information by inputting the image of the grid and the real-world distances between the dots into the IMAQ Learn Calibration Template VI (**Vision Utilities»Calibration**).

Note If you want to specify a list of points instead of a grid, use the **Reference Points** control of IMAQ Learn Calibration Template to specify the pixel to real-world mapping.

Specifying Scaling Factors

Scaling factors are the real-world distances between the dots in the calibration grid in the x and y directions and the units in which the distances are measured. Use the **X Step** and **Y Step** elements of the **Grid Descriptor** control to specify the scaling factors.

Choosing a Region of Interest

Define a learning ROI during the learning process to define a region of the calibration grid you want to learn. The software ignores dot centers outside this region when it estimates the transformation. Depending on the calibration options selected, this is an effective way to increase correction speeds. Set the learning ROI using the **Calibration Learn Setup** control of IMAQ Learn Calibration Template.

Note The user-defined learning ROI represents the area in which you are interested. Do not confuse the learning ROI with the calibration ROI generated by the calibration algorithm. Refer to Figure 6-6 for an illustration of calibration ROIs.

Choosing a Learning Algorithm

Select a method in which to learn the calibration information: perspective projection or nonlinear. Figure 6-5 illustrates the types of errors your image can exhibit. Figure 6-5a shows an image of a calibration grid with no errors. Figure 6-5b shows an image of a calibration grid with perspective projection. Figure 6-5c shows an image of a calibration grid with nonlinear distortion.

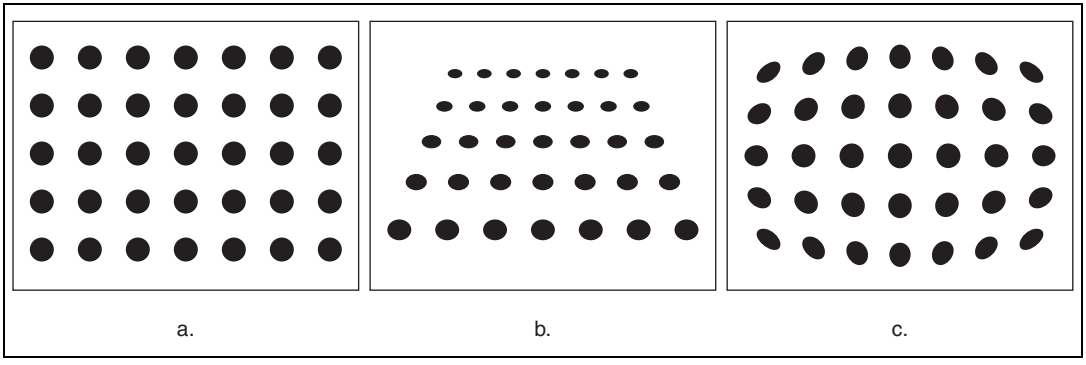
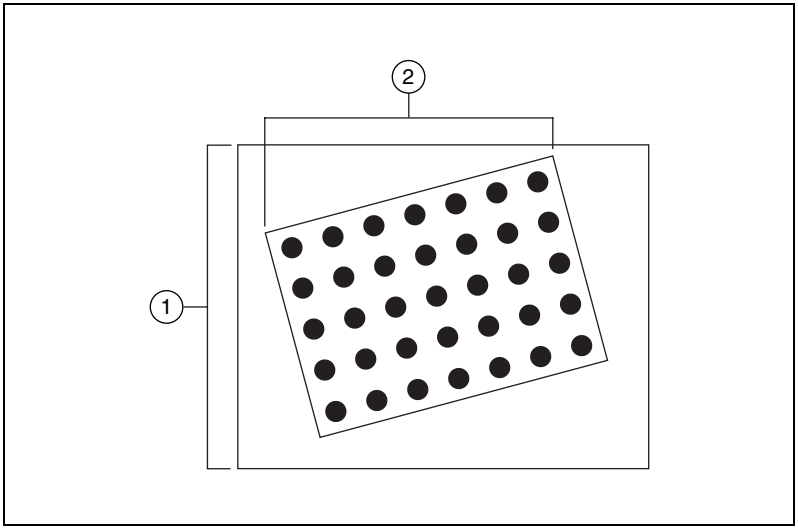


Figure 6-5. Types of Image Distortion

Choose the perspective projection algorithm when your system exhibits perspective errors only. A perspective projection calibration has an accurate transformation even in areas not covered by the calibration grid, as shown in Figure 6-6. Set the **Distortion** element of the **Calibration Learn Setup** control to **Perspective** to choose the perspective calibration algorithm. Learning and applying perspective projection is less computationally intensive than the nonlinear method. However, perspective projection is not designed to handle highly nonlinear distortions.

If your imaging setup exhibits nonlinear distortion, use the nonlinear method. The nonlinear method guarantees accurate results only in the area that the calibration grid covers, as shown in Figure 6-6. If your system exhibits both perspective and nonlinear distortion, use the nonlinear method to correct for both. Set the **Distortion** element of the **Calibration Learn Setup** control to **Nonlinear** to choose the nonlinear calibration algorithm.



1	Calibration ROI Using the Perspective Algorithm	2	Calibration ROI Using the Nonlinear Algorithm

Figure 6-6. Calibration ROIs

Using the Learning Score

The learning process returns a score that reflects how well the software learned the input image. A learning score above 800 indicates that you chose the appropriate learning algorithm, that the grid image complies with the guideline, and that your vision system setup is adequate.

If the learning process returns a learning score below 600, try the following:

1. Make sure your grid complies with the guidelines listed in the *Defining a Calibration Template* section of this chapter.
2. Check the lighting conditions. If you have too much or too little lighting, the software may estimate the center of the dots incorrectly. Also, adjust your **Threshold Range** to distinguish the dots from the background.
3. Select another learning algorithm. When nonlinear lens distortion is present, using perspective projection sometimes results in a low learning score.

Note A high score does *not* reflect the accuracy of your system.

Learning the Error Map

An error map helps you gauge the quality of your complete system. The error map returns an estimated error range to expect when a pixel coordinate is transformed into a real-world coordinate. The transformation accuracy may be higher than the value the error range indicates. Set the **Learn Error Map?** element of the **Calibration Learn Setup** control to learn the error map.

Learning the Correction Table

If the speed of image correction is a critical factor for your application, use a correction table. A correction table is a lookup table, stored in memory, that contains the real-world location information of all the pixels in the image. The extra memory requirements for this option are based on the size of the image. Use this option when you want to correct several images at a time in your vision application. Set the **Learn Correction Table?** element of the **Calibration Learn Setup** control to learn the correction table.

Setting the Scaling Method

Use the **Corrected Image Scaling** element of the **Calibration Learn Setup** control to choose the appearance of the corrected image. Select either **Scale to Fit** or **Scale to Preserve Area**. Refer to Chapter 3, *System Setup and Calibration*, of the *NI Vision Concepts Manual* for more information about the scaling mode.

Calibration Invalidation

Any image processing operation that changes the image size or orientation voids the calibration information in a calibrated image. Examples of VIs that void calibration information include IMAQ Resample, IMAQ Extract, IMAQ ArrayToImage, and IMAQ Unwrap.

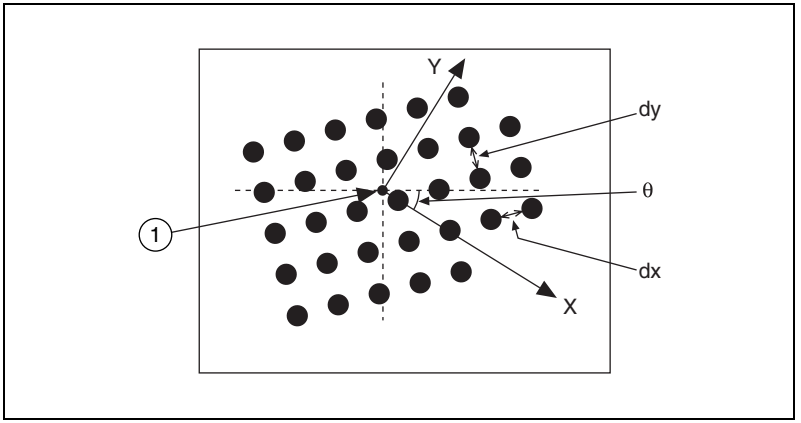
Simple Calibration

When the axis of your camera is perpendicular to the image plane and lens distortion is negligible, use a simple calibration. In a simple calibration, a pixel coordinate is transformed to a real-world coordinate through scaling in the horizontal and vertical directions.

Use simple calibration to map pixel coordinates to real-world coordinates directly without a calibration grid. The software rotates and scales a pixel coordinate according to predefined coordinate reference and scaling factors. You can assign the calibration to an arbitrary image using the IMAQ Set Simple Calibration VI (**Vision Utilities»Calibration**).

To perform a simple calibration, set a coordinate reference—composed of an angle, center, and axis direction—and scaling factors on the defined axis, as shown in Figure 6-7. Express the angle between the x-axis and the horizontal axis of the image in degrees. Express the center as the position, in pixels, where you want the coordinate reference origin. Set the axis direction to direct or indirect. Simple calibration also offers a correction table option and a scaling mode option.

Simple calibration is performed using IMAQ Set Simple Calibration. Use the **Calibration Axis Info** control to define the coordinate system. Use the **X Step** and **Y Step** elements of the **Grid Descriptor** control to specify the scaling factors. Use the **Corrected Image Scaling** control to set the scaling method. Set the **Learn Correction Table?** control to **True** to learn the correction table.



1 Origin

Figure 6-7. Defining a Simple Calibration

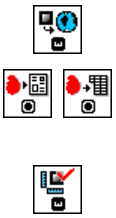
Save Calibration Information

After you learn the calibration information, you can save it so that you do not have to relearn the information for subsequent processing. Use the IMAQ Write Image and Vision Info VI (**Vision Utilities»Calibration**) to save the image of the grid and its associated calibration information to a file. To read the file containing the calibration information use the IMAQ Read Image and Vision Info VI (**Vision Utilities»Calibration**). Refer to the *Attach Calibration Information* section of this chapter for more information about attaching the calibration information you read from another image.

Attach Calibration Information

Now that you have calibrated your setup correctly, you can apply the calibration settings to images that you acquire. Use the IMAQ Set Calibration Info VI (**Vision Utilities»Calibration**) to attach the calibration information of the current setup to each image you acquire. This VI takes in a source image containing the calibration information and a destination image that you want to calibrate. The destination image is your inspection image with the calibration information attached to it.

Note The source image and destination image must be the same size.



Using the calibration information attached to the image, you can accurately convert pixel coordinates to real-world coordinates to make any of the analytic geometry measurements with IMAQ Convert Pixel to Real World (**Vision Utilities»Calibration**). If your application requires that you make shape measurements, you can use the calibrated measurements from the IMAQ Particle Analysis or IMAQ Particle Analysis Report VIs (**Image Processing»Analysis**). You also can correct the image by removing distortion with IMAQ Correct Calibrated Image.

Note Correcting images is a time-intensive operation.

A calibrated image is not the same as a corrected image. Because calibration information is part of the image, it is propagated throughout the processing and analysis of the image. Functions that modify the image size, such as an image rotation function, void the calibration information. Use IMAQ Write Image and Vision Info (**Vision Utilities»Calibration**) to save the image and all of the attached calibration information to a file.

A

Using NI Vision with the LabVIEW Real-Time Module

This appendix introduces the real-time capabilities of NI Vision and describes how you can use NI Vision with the LabVIEW Real-Time Module to create a machine vision application for a real-time, deterministic, embedded target.

Overview

With the LabVIEW Real-Time Module, NI-IMAQ, and NI Vision, you have all the tools necessary to develop a complete machine vision application on a reliable, embedded platform. The LabVIEW Real-Time Module provides real-time programming and execution capabilities, NI-IMAQ provides the acquisition components, and NI Vision provides the image manipulation and analysis functions.

Develop your vision application with NI-IMAQ and NI Vision. Then download your code to run on a real-time, embedded target. You also can add National Instruments data acquisition (*DAQ*), motion control, contoller area network (*CAN*), and serial instruments to your LabVIEW Real-Time Module system to create a complete, integrated, embedded system.

System Components

If you are using NI Vision with the LabVIEW Real-Time Module, your system consists of a development system and one or more deployed RT targets.

Development System

The development system for NI Vision and the LabVIEW Real-Time Module is made up of the following major components:

- Host—Pentium-based machine running a Windows operating system. Use this component to configure your NI PXI controller or

NI CVS-1450 Series device as an RT target and to develop your application.

- RT target—RT Series hardware that runs VIs downloaded from and built in LabVIEW. Examples of RT targets include a PXI chassis housing a PXI controller or a CVS-1450 Series device.

Refer to the *NI Vision Hardware Help* for details about how to set up each machine and how they interact with each other.

Note You need a network connection between your host machine and RT target during development to configure the RT target and to download software and code from your host machine onto the RT target. This network connection is optional at runtime.

Deployed System

When you have configured your host development system, you can set up and configure additional RT targets for deployment. These deployed systems use the same hardware and software as your development RT target, but they do not require Windows for configuration. Instead of using Windows for configuration, copy your configuration information from your development RT target to the deployment system.

Software Installation

Set up your RT target by installing the LabVIEW Real-Time Module and NI-IMAQ. Refer to the *NI Vision Hardware Help* for detailed instructions.

Use MAX to install NI Vision and any other necessary LabVIEW Real-Time Module components from your host machine onto your RT target system. Refer to the *Measurement & Automation Explorer Remote Systems Help* for specific information (within MAX, go to **Help»Help Topics»Remote Systems**).

When your RT target is set up, you can write and execute NI Vision code just as you would on a Windows-based system.

Image Display

Using NI Vision with the LabVIEW Real-Time Module gives you two options for displaying images: Remote Display and RT Video Out. Use Remote Display during development and debugging to view your images from your host machine just as you would view the LabVIEW front panels of the VIs running on your LabVIEW Real-Time Module system. Use RT

Video Out to display your images on a monitor connected to your remote LabVIEW Real-Time Module system.

Remote Display

Remote Display allows you to acquire images on your remote system and view them on your host machine. Remote Display is automatically enabled when you use the LabVIEW Image Display control (available with LabVIEW 7.0 or later) or any of the NI Vision display VIs (**Vision Utilities»External Display**)—such as IMAQ WindDraw, IMAQ WindToolsShow, and IMAQ ConstructROI.

Remote Display is useful for monitoring and debugging your applications that use NI Vision with the LabVIEW Real-Time Module. Familiarize yourself with how Remote Display works before using this feature.

The following details will help you prepare your application for use with Remote Display:

- Remote Display is a front-panel feature. Therefore, your LabVIEW front-panel must be open for you to see images displayed using Remote Display.
- Remote Display performs best when combined with IMAQ Remote Display Options. (**Vision Utilities»Vision RT**). When you display images on your remote machine, LabVIEW must send those images over your network. This process can take up a large amount of your network bandwidth, especially when transferring large images. IMAQ Remote Display Options allows you to specify compression settings for those images to reduce the network bandwidth used by the display process. In addition, compressing images may increase your display rates on slower networks.

Note IMAQ Remote Display Options will not affect the remote display options in versions of LabVIEW prior to LabVIEW 8.0. Use the IMAQ Flatten Image Options VI instead.

- IMAQ Remote Display Options uses two types of compression algorithms. Use the lossy JPEG compression algorithm on grayscale and color images. Use the lossless Packed Binary compression algorithm on binary images. Refer to the *NI Vision for LabVIEW VI Reference Help* for more information about the IMAQ Remote Display Options VI.

Note JPEG Compression may result in data degradation of the displayed image. There is no degradation of the image during processing. Test various compression settings to find the right balance for your application.

- Using Remote Display can affect the timing performance of your NI Vision VIs. Do not use Remote Display if your program contains a time-critical loop.
- Disconnecting your remote system from your host machine disables Remote Display. Disabled Remote Display VIs do not affect the performance of your application. When you reconnect your remote system and host machine, Remote Display is automatically re-enabled.
- Use RT Video Out instead of Remote Display on deployed systems. Remote Display requires a LabVIEW front panel, and deployed systems do not have a front panel. Refer to the *LabVIEW Help* for recommended program architecture.

Tip Refer to the *Remote Display Errors* section of this appendix for more information.

RT Video Out

RT Video Out allows you to display images on a monitor that is connected to your RT target. In NI Vision, IMAQ WindDraw and IMAQ WindClose (**Vision Utilities»External Display**) provide support for RT Video Out. To display images on a monitor connected to your RT target, input `15` for the **Window Number** control.

Alternately, you can use IMAQ RT Video Out (**Vision Utilities» Vision RT**) to display image on a monitor connected to your RT target.

Note This feature is available only on controllers that feature the i815 chipset, such as NI PXI-8175/76 Series controllers and NI CVS-1450 Series compact vision system.

RT Video Out supports overlay functionality. However, the overlay text is limited to one size and one font.

This display option is not a time-bounded operation. Refer to the *Determinism in Real-Time Applications* section of this appendix for more information about time-bounded operations.

To programmatically configure your system to use RT Video Out for displaying system images, use the IMAQ Video Out Display Mode VI (**Vision Utilities»Vision RT**). This VI allows you to set parameters for screen area, color depth, and refresh frequency.

Determinism in Real-Time Applications

An algorithm exhibits *determinism* when the time to execute the algorithm is predictable and repeatable given any valid input set. Executing a deterministic algorithm takes a predetermined amount of time within a prescribed variance.

Determinism is a product of both the algorithm and the system on which it executes. Real-time systems, such as the LabVIEW Real-Time Module, provide the foundation for you to deterministically execute algorithms. However, determinism is not guaranteed without resource management in any real-time system that has dynamically controlled resources. Resource contention—the inability of a process to access needed resources immediately during execution—destroys determinism. Examples of resource contention in a vision application include the following:

- Parallel processes that use the same image
- Vision algorithms that allocate memory, even for internal workspace, from the systems memory manager

Resource contention is just one example of how to destroy determinism. Determinism also can be destroyed by the nature of some algorithms, by adding file I/O, or by networking. Refer to the *LabVIEW Help* for a discussion on determinism and programming deterministic applications.

Determinism versus Time-Bounded Execution

Due to the complexity of vision algorithms and the dramatic variance in their input sets (mostly images), determinism, as defined in the previous section, is not attainable. For example, to be deterministic, a pattern matching routine would have to produce results in the same amount of time for any template on any image regardless of content or size.

However, one of the most important characteristics of determinism, limiting execution time, is achievable as long as a final result is not expected for any given time limit. The deterministic condition requires both a lower and upper bound on execution time. Algorithms designed to support execution caps are referred to as *time-bounded*. When time is exceeded, these algorithms return with a timeout error.

For certain vision algorithms, the execution time has relatively small *jitter* if the input sets are similar. For instance, pattern matching produces results in roughly the same time when searching for the same pattern in images with common content. Therefore, many vision applications already contain components that have consistent execution times. Running the application

on the LabVIEW Real-Time Module enhances the time reliability. Unfortunately, this execution behavior is dependant on the commonality of the input sets. In many applications, the input sets are common enough that you can safely predict the execution time by running the application over a large, representative set of example images. In some cases, however, getting a representative set of example images may be difficult, or a bounded execution time may be particularly important. Herein lies the need for time-bounded algorithms.

Time-Bounded Execution

As with determinism, time-bounded behavior is controlled by both the algorithm and the environment in which it executes. For this reason, some vision algorithms are not candidates for time bounding. For example, algorithms whose execution time varies widely between similar images are not productive under time constraints. This includes operations, such as skeleton, separation, and segmentation, whose execution time can vary dramatically with slight changes in the input image. This makes choosing a time limit for such operations difficult. However, many vision algorithms are adaptable to time limits when the appropriate timed environment is established.

When using NI Vision with the LabVIEW Real-Time Module, the timed environment is best described in terms of the following process flow:

1. Initialize the timed environment.
2. Prepare resources.
3. Perform time-bounded vision operations.
4. Close the timed environment.

Initializing the Timed Environment

You must initialize the timed environment to manage resource allocation. While the LabVIEW Real-Time Module manages all of the resources used by the vision application, some resources must be allocated dynamically, which leaves the possibility for resource contention. For example, if a vision algorithm needs to allocate memory for internal workspace, it may contend with another piece in LabVIEW for access to the memory manager. This could cause it to lose priority to another execution thread. To alleviate this problem, pre-allocate resources during initialization for internal algorithm workspace and the timing mechanism. Use the IMAQ Initialize Timed Execution VI (**Vision Utilities»Vision RT**) to preallocate memory used by the NI Vision algorithms.

Use IMAQ Initialize Timed Execution to initialize the environment. Because resource requirements differ among vision applications, you can change the amount of memory reserved using the **Reserved Memory Size** control. If the reserved resources are exhausted during execution, a special out-of-memory error message is generated. If you receive this error, increase the amount of resource memory to meet the needs of your processing.

The resources you reserve at initialization are not used until the timing mechanism is started. These resources are intended for use in internal processing that is not exposed in the LabVIEW environment. For objects that are exposed in LabVIEW, always preallocate resources before entering the time-bounded portion of your code. For example, preallocate destination images using IMAQ Create (**Vision Utilities»Image Management**) and IMAQ SetImageSize (**Vision Utilities»Image Management**) before entering time-bounded code.

Preparing Resources

Allocate any resource whose exact size you know before the time limit is started. This encourages optimal use of the reserved resources and provides maximum flexibility.

System resources allocated before timed execution are available at any time. Reserved resources allocated inside the time-bounded portion of your code are not guaranteed to be available outside the timed environment. Therefore, you should preallocate as much as possible before entering the time-bounded portion of your code. When time-bounded execution begins, changes to system resources by NI Vision algorithms, such as resizing an existing image, generate an error.

Images can be created only in system resources. In addition, special image changes performed by learning a pattern, calibrating an image, or adding an overlay also require system resources. This is primarily because they have to exist outside the timed environment. These operations are prohibited during time-bounded execution.

Performing Time-Bounded Vision Operations

To limit the execution time of a block of vision operations, make the following modifications:

- Isolate the portion of code to time-bound in an independent VI.
- Set the independent VI's execution priority to time-critical so that lower priority processes do not interrupt the timed execution. Refer to the *LabVIEW Help* for information about program architecture.
- Turn off automated error handling so that dialog boxes are not generated when an error occurs.
- Replace all Vision VIs in the time-bounded operation with their time-bounded counterparts located in `Preallocated.llb`. Using a Vision VI that is not time-bounded generates a run-time error if the time limit is set. Replacing the VIs ensures that Vision algorithms do not request resources from the system while running.

- Call IMAQ Start Timed Execution (**Vision Utilities»Vision RT**) at the beginning of the vision block to initiate the time limit. Use IMAQ Stop Timed Execution (**Vision Utilities»Vision RT**) to turn off the time limit at the end of the Vision block. Connect the error output from IMAQ Start Timed Execution to the Vision VIs in sequential order ending with IMAQ Stop Timed Execution. When time expires and the processing is not complete, the Vision VI executing at that moment generates a special timeout error. The error cluster propagates the timeout condition reducing overall execution jitter. Attempting to start an additional time limit results in an error.

For vision algorithms working with images, serialized processing is crucial because an image may be shared among multiple vision routines running in parallel. When changes to the image are required, the image is blocked from access until the updates are completed. This is another form of resource contention that invalidates time constraints. If the error cluster is passed between VIs sequentially, this type of conflict is avoided. Use the error cluster to sequence your VIs and one loop in the code to avoid time constraint conflicts.

All non-vision processing during time-bounded execution is not constrained by the timing mechanism, increasing the jitter of the overall execution. Consequently, limit non-vision processing during time-bounded execution. In particular, eliminate any operation in the LabVIEW Real-Time Module requesting resources because these operations nullify the time limit. For example, do not build or resize arrays to sizes determined at run-time if a time limit is active. However, you can perform such operations as reading elements from an array. Other operations, such

as file I/O, serial I/O, and networking, are inherently non-deterministic and should be avoided in the time-critical portion of your application. Refer to the LabVIEW Real-Time Module documentation to determine which routines can execute deterministically.

Because some non-vision processing may be required during timed execution, use IMAQ Check Timed Execution (**Vision Utilities» Vision RT**) periodically to see if time has expired. Determine how frequently you need to check for expired time based on the complexity of the non-vision process and the required jitter.

Closing the Timed Environment

When time-bounded execution is no longer needed, call IMAQ Uninitialize Timed Execution (**Vision Utilities»Vision RT**) to release the resources reserved at initialization. When the environment is uninitialized, calls to IMAQ Start Timed Execution produce an error.

Image Files

Many applications require you to use external files, such as the template files used by pattern matching and spatial calibration functions. Before running your application on an RT target, you must use FTP to transfer any external image files from your host machine to your remote target. You can use MAX 3.0 and later to FTP images to your RT target. Refer to the *LabVIEW Help* for more information about using FTP.

Deployment

When you have finished developing your application using NI Vision for the LabVIEW and the LabVIEW Real-Time Module, you may want to deploy that application to a number of remote systems. To achieve consistent results from your application, you must configure these deployed systems with the same settings you used for your development system.

Note Each deployed system must have its own embedded real-time controller and software. Visit ni.com for ordering information.

Note You must purchase a run-time license for NI Vision and a run-time license for the LabVIEW Real-Time Module for each deployed system using NI Vision and the LabVIEW Real-Time Module. Visit ni.com for more information about purchasing run-time licenses.

Troubleshooting

This section describes solutions and suggestions for common errors when using NI Vision with the LabVIEW Real-Time Module.

Remote Display Errors

Why am I unable to display remote images using IMAQ WindDraw?

The Vision Remote Server controls the displaying of images acquired from your remote systems in external windows. If you are unable to display your remote images, press <Ctrl-Alt-Del>, which opens the Windows Task Manager, and confirm that `NIVisSvr.exe` is running as an active process. If `NIVisSvr.exe` is not running, close and restart the LabVIEW Real-Time Module and any open NI Vision applications.

If you still experience errors, contact National Instruments Technical Support. Technical support information is available in Appendix B, *Technical Support and Professional Services*.

Why does my remotely displayed image have low quality?

Try these steps to improve your image quality:

- Ensure that your camera aperture is open to allow the appropriate amount of light for an acquisition.
- Check your compression settings.
- Make sure that your display settings in Windows are set to use at least 24-bit color.

How do I select an ROI in an image?

If you are using the LabVIEW Real-Time Module 7.0 or later, you can you use an Image Display control to select an ROI. Wire your inspection image to the Image Display control on your LabVIEW block diagram to display the image on your LabVIEW front panel.

While your application is running on the RT target, your user can use the ROI tools associated with the Image Display control to select their ROI on the host computer. Use the ROI property node in your application to get the user selected ROI.

If you are using the LabVIEW Real-Time Module 7.1 or later, you can use the Get Last Event invoke node in your application to determine when a user has drawn a new ROI and to get basic properties of the new ROI. You can also use the Clear ROI invoke node to ensure the Image Display control

is not displaying an ROI when your application begins execution. Refer to `Extract Example.vi` located in `examples\vision\2. Functions\Image Management` to learn more about using these invoke nodes in an application.

For all versions of LabVIEW RT, you can use the IMAQ Construct ROI VI (**Vision Utilities»External Display**) to allow the user to select an ROI. LabVIEW displays the ROI selection window on the host computer and outputs the selected ROI to the VI running on the RT target. Similarly, you can use the IMAQ WindDraw VI (**Vision Utilities»External Display**) to display the inspection image on the host computer. You can then use the IMAQ WindToolsShow VI (**Vision Utilities»External Display**) to display the tools dialog on the host computer. Your user would then use the ROI tools to select their ROI in the image display window. Use the IMAQ WindGetROI VI (**Vision Utilities»External Display**) to get the user selected ROI for processing on the RT target.

Programming Errors

Why won't my LabVIEW VI run on my RT target?

Your NI Vision VI may not be supported by the LabVIEW Real-Time Module. The following VIs are among those not supported:

- IMAQ Browser Delete (**Vision Utilities»External Display»Browser**)

- IMAQ Browser Focus (**Vision Utilities»External Display»Browser**)

- IMAQ Browser Focus Setup (**Vision Utilities»External Display»Browser**)

- IMAQ Browser Insert (**Vision Utilities»External Display»Browser**)

- IMAQ Browser Replace (**Vision Utilities»External Display»Browser**)

- IMAQ Browser Setup (**Vision Utilities»External Display»Browser**)

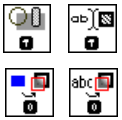
- IMAQ Draw (**Vision Utilities»Pixel Manipulation**)

- IMAQ Draw Text (**Vision Utilities»Pixel Manipulation**)

- IMAQ WindGrid (**Vision Utilities»External Display**)

- All IMAQ AVI VIs (**Vision Utilities»Files»AVI**)
- Other obsoleted VIs. If your program contains a VI that has been updated or replaced to support new functionality, the icon of the obsoleted VI will contain a small black *X*.

How can I make my NI Vision application work on my LabVIEW Real-Time Module system if it contains IMAQ Draw or IMAQ Draw Text?



The IMAQ Draw and IMAQ Draw Text VIs are not supported under the LabVIEW Real-Time Module. However, you can achieve similar functionality by using the IMAQ Overlay VIs (**Vision Utilities»Overlays**), such as IMAQ Overlay Rectangle and IMAQ Overlay Text. You also can use IMAQ Merge Overlay to merge your overlay data into your image. Refer to the *NI Vision for LabVIEW VI Reference Help* for information about merging overlays.

How can I make my NI Vision application work on my LabVIEW Real-Time Module system if the application contains IMAQ Browser VIs?

If your application uses any of the IMAQ Browser VIs, use IMAQ ImageToImage (**Vision Utilities»Image Management**) to embed multiple images within a single image.

Why do I get a `File Not Found` error from my LabVIEW VI when I run it on the LabVIEW Real-Time Module system?

When you run your LabVIEW Real-Time Module application, your VI is downloaded to your RT target, but your support files—such as images and templates—are not. The File I/O routines in the LabVIEW Real-Time Module and NI Vision always refer to files on the target machine, which is the remote RT target in this case. Use FTP to move your image files to the RT target.

Refer to your LabVIEW Real-Time Module documentation for more information about transferring files to your RT target using FTP.

If you created a VI in the NI Vision Assistant, selecting **Image File** for your **Image Source** setting causes your VI to return an error in the LabVIEW Real-Time Module. This is because the File Dialog function is not supported in the LabVIEW Real-Time Module. To avoid this error, select **Image Control** or **Image Acquisition Board** as the Image Source.

RT Video Out Errors

Why do I have an invalid Video Out Mode?

To use the RT Video Out functionality with NI Vision and the LabVIEW Real-Time Module, you must have a PXI controller featuring the i815 graphics chipset, such as the National Instruments PXI 8175/6 Series controllers.

If you are using a controller that does not support RT Video Out, consider using Remote Display to display your images.

Why can't I see my images when I use RT Video Out?

Use the IMAQ Video Out Display Mode VI to configure your video mode before you attempt to display your images. This VI allows you to set your refresh frequency, screen area, and color depth.

Note If you are using a monitor that does not support high refresh frequencies, your images cannot display correctly. Refer to your monitor documentation for information about supported refresh frequencies.

B

Technical Support and Professional Services

Visit the following sections of the National Instruments Web site at ni.com for technical support and professional services:

- **Support**—Online technical support resources at ni.com/support include the following:
  - **Self-Help Resources**—For answers and solutions, visit the award-winning National Instruments Web site for software drivers and updates, a searchable KnowledgeBase, product manuals, step-by-step troubleshooting wizards, thousands of example programs, tutorials, application notes, instrument drivers, and so on.
  - **Free Technical Support**—All registered users receive free Basic Service, which includes access to hundreds of Application Engineers worldwide in the NI Developer Exchange at ni.com/exchange. National Instruments Application Engineers make sure every question receives an answer.

    For information about other technical support options in your area, visit ni.com/services or contact your local office at ni.com/contact.
- **Training and Certification**—Visit ni.com/training for self-paced training, eLearning virtual classrooms, interactive CDs, and Certification program information. You also can register for instructor-led, hands-on courses at locations around the world.
- **System Integration**—If you have time constraints, limited in-house technical resources, or other project challenges, National Instruments Alliance Partner members can help. To learn more, call your local NI office or visit ni.com/alliance.

If you searched ni.com and could not find the answers you need, contact your local office or NI corporate headquarters. Phone numbers for our worldwide offices are listed at the front of this manual. You also can visit the Worldwide Offices section of ni.com/niglobal to access the branch office Web sites, which provide up-to-date contact information, support phone numbers, email addresses, and current events.

Glossary

Numbers

1D	One-dimensional.
2D	Two-dimensional.
3D	Three-dimensional.

A

AIPD	The National Instruments internal image file format used for saving complex images and calibration information associated with an image. AIPD images have the file extension *APD*.
alignment	The process by which a machine vision application determines the location, orientation, and scale of a part being inspected.
alpha channel	The channel used to code extra information, such as gamma correction, about a color image. The alpha channel is stored as the first byte in the four-byte representation of an RGB pixel.
area	(1) A rectangular portion of an acquisition window or frame that is controlled and defined by software. (2) The size of an object in pixels or user-defined units.
arithmetic operators	The image operations multiply, divide, add, subtract, and modulo.
array	An ordered, indexed set of data elements of the same type.
auto-median function	A function that uses dual combinations of opening and closing operations to smooth the boundaries of objects.

B

b	Bit. One binary digit, either 0 or 1.
B	Byte. Eight related bits of data, an 8-bit binary number. Also denotes the amount of memory required to store one byte of data.
barycenter	The grayscale value representing the centroid of the range of an image's grayscale values in the image histogram.
binary image	An image in which the objects usually have a pixel intensity of 1 (or 255) and the background has a pixel intensity of 0.
binary morphology	Functions that perform morphological operations on a binary image.
binary threshold	The separation of an image into objects of interest (assigned pixel values of 1) and background (assigned pixel values of 0) based on the intensities of the image pixels.
bit depth	The number of bits (n) used to encode the value of a pixel. For a given n, a pixel can take 2^n different values. For example, if n equals 8, a pixel can take 256 different values ranging from 0 to 255. If n equals 16, a pixel can take 65,536 different values ranging from 0 to 65,535 or –32,768 to 32,767.
blurring	Reduces the amount of detail in an image. Blurring can occur when the camera lens is out of focus or when an object moves rapidly in the field of view. You can blur an image intentionally by applying a lowpass frequency filter.
BMP	Bitmap. An image file format commonly used for 8-bit and color images. BMP images have the file extension *BMP*.
border function	Removes objects (or particles) in a binary image that touch the image border.
brightness	(1) A constant added to the red, green, and blue components of a color pixel during the color decoding process. (2) The perception by which white objects are distinguished from gray and light objects from dark objects.
buffer	Temporary storage for acquired data.

C

caliper — A measurement function that finds edge pairs along a specified path in the image. This function performs an edge extraction and then finds edge pairs based on specified criteria such as the distance between the leading and trailing edges, edge contrasts, and so forth.

CAN — Controller Area Network. A serial bus finding increasing use as a device-level network for industrial automation. CAN was developed by Bosch to address the needs of in-vehicle automotive communications.

center of mass — The point on an object where all the mass of the object could be concentrated without changing the first moment of the object about any axis.

chroma — The color information in a video signal.

chromaticity — The combination of hue and saturation. The relationship between chromaticity and brightness characterizes a color.

closing — A dilation followed by an erosion. A closing fills small holes in objects and smooths the boundaries of objects.

clustering — A technique where the image is sorted within a discrete number of classes corresponding to the number of phases perceived in an image. The gray values and a barycenter are determined for each class. This process is repeated until a value is obtained that represents the center of mass for each phase or class.

CLUT — Color lookup table. A table for converting the value of a pixel in an image into a red, green, and blue (RGB) intensity.

color image — An image containing color information, usually encoded in the RGB form.

color space — The mathematical representation for a color. For example, color can be described in terms of red, green, and blue; hue, saturation, and luminance; or hue, saturation, and intensity.

complex image — Stores information obtained from the FFT of an image. The complex numbers that compose the FFT plane are encoded in 64-bit floating-point values: 32 bits for the real part and 32 bits for the imaginary part.

connectivity — Defines which of the surrounding pixels of a given pixel constitute its neighborhood.

connectivity-4	Only pixels adjacent in the horizontal and vertical directions are considered neighbors.
connectivity-8	All adjacent pixels are considered neighbors.
contrast	A constant multiplication factor applied to the luma and chroma components of a color pixel in the color decoding process.
convex hull	The smallest convex polygon that can encapsulate a particle.
convex hull function	Computes the convex hull of objects in a binary image.
convolution	*See* linear filter.
convolution kernel	2D matrices, or templates, used to represent the filter in the filtering process. The contents of these kernels are a discrete 2D representation of the impulse response of the filter that they represent.
curve extraction	The process of finding curves, or connected edge points, in a grayscale image. Curves usually represent the boundaries of objects in the image.

D

Danielsson function	Similar to the distance functions, but with more accurate results.
DAQ	Data acquisition. The process of collecting and measuring electrical signals from sensors, transducers, and test probes or fixtures and inputting them to a computer for processing.
determinism	A characteristic of a system that describes how consistently it can respond to external events or perform operations within a given time limit.
digital image	An image $f(x, y)$ that has been converted into a discrete number of pixels. Both spatial coordinates and brightness are specified.
dilation	Increases the size of an object along its boundary and removes tiny holes in the object.
driver	Software that controls a specific hardware device, such as an NI image acquisition or DAQ device.

E

edge	Defined by a sharp transition in the pixel intensities in an image or along an array of pixels.
edge contrast	The difference between the average pixel intensity before the edge and the average pixel intensity after the edge.
edge detection	Any of several techniques that identify the edges of objects in an image.
edge steepness	The number of pixels that corresponds to the slope or transition area of an edge.
energy center	The center of mass of a grayscale image. *See* center of mass.
equalize function	*See* histogram equalization.
erosion	Reduces the size of an object along its boundary and eliminates isolated points in the image.
exponential and gamma corrections	Expand the high gray-level information in an image while suppressing low gray-level information.
exponential function	Decreases brightness and increases contrast in bright regions of an image, and decreases contrast in dark regions of an image.

F

FFT	Fast Fourier Transform. A method used to compute the Fourier transform of an image.
fiducial	A reference pattern on a part that helps a machine vision application find the part location and orientation in an image.
Fourier transform	Transforms an image from the spatial domain to the frequency domain.
frequency filters	The counterparts of spatial filters in the frequency domain. For images, frequency information is in the form of spatial frequency.
function	A set of software instructions executed by a single line of code that may have input and/or output parameters and returns a value when executed.

G

gamma	The nonlinear change in the difference between the video signal's brightness level and the voltage level needed to produce that brightness.
geometric features	Information extracted from a grayscale template that are used to locate the template in the target image. Geometric features in an image range from low-level features such as edges or curves detected in the image to higher-level features such as the geometric shapes made by the curves in the image.
geometric matching	A technique used to locate quickly a grayscale template that is characterized by distinct geometric or shape information within a grayscale image.
Golden Template comparison	A comparison of the pixel intensities of an image under inspection to a golden template. A golden template is an image containing an ideal representation of an object under inspection.
gradient convolution filter	*See* gradient filter.
gradient filter	An edge detection algorithm that extracts the contours in gray-level values. Gradient filters include the Prewitt and Sobel filters.
gray level	The brightness of a pixel in an image.
gray-level dilation	Increases the brightness of pixels in an image that are surrounded by other pixels with a higher intensity.
gray-level erosion	Reduces the brightness of pixels in an image that are surrounded by other pixels with a lower intensity.
grayscale image	An image with monochrome information.
grayscale morphology	Functions that perform morphological operations on a gray-level image.

H

highpass attenuation	The inverse of lowpass attenuation.
highpass filter	Emphasizes the intensity variations in an image, detects edges or object boundaries, and enhances fine details in an image.
highpass frequency filter	Removes or attenuates low frequencies present in the frequency domain of the image. A highpass frequency filter suppresses information related to slow variations of light intensities in the spatial image.
highpass truncation	The inverse of lowpass truncation.
histogram	Indicates the quantitative distribution of the pixels of an image per gray-level value.
histogram equalization	Transforms the gray-level values of the pixels of an image to occupy the entire range of the histogram, thus increasing the contrast of the image. The histogram range in an 8-bit image is 0 to 255.
histogram inversion	Finds the photometric negative of an image. The histogram of a reversed image is equal to the original histogram flipped horizontally around the center of the histogram.
histograph	In LabVIEW, a histogram that can be wired directly into a graph.
hit-miss function	Locates objects in the image similar to the pattern defined in the structuring element.
HSI	A color encoding scheme in hue, saturation, and intensity.
HSL	A color encoding scheme using hue, saturation, and luminance information where each image in the pixel is encoded using 32 bits: eight bits for hue, eight bits for saturation, eight bits for luminance, and eight unused bits.
HSV	A color encoding scheme in hue, saturation, and value.
hue	Represents the dominant color of a pixel. The hue function is a continuous function that covers all the possible colors generated using the R, G, and B primaries. *See also* RGB.
Hz	Hertz. Frequency in units of 1/second.

I

I/O	Input/output. The transfer of data to/from a computer system involving communications channels, operator interface devices, and/or data acquisition and control interfaces.
image	A 2D light intensity function $f(x, y)$ where x and y denote spatial coordinates and the value f at any point (x, y) is proportional to the brightness at that point.
image border	A user-defined region of pixels surrounding an image. Functions that process pixels based on the value of the pixel neighbors require image borders.
Image Browser	An image that contains thumbnails of images to analyze or process in a vision application.
image buffer	A memory location used to store images.
image definition	The number of values a pixel can take on, which is the number of colors or shades that you can see in the image.
image display environment	A window or control that displays an image.
image enhancement	The process of improving the quality of an image that you acquire from a sensor in terms of signal-to-noise ratio, image contrast, edge definition, and so on.
image file	A file containing pixel data and additional information about the image.
image format	Defines how an image is stored in a file. Usually composed of a header followed by the pixel data.
image mask	A binary image that isolates parts of a source image for further processing. A pixel in the source image is processed if its corresponding mask pixel has a nonzero value. A source pixel whose corresponding mask pixel has a value of 0 is left unchanged.
image palette	The gradation of colors used to display an image on screen, usually defined by a CLUT.
image processing	Encompasses various processes and analysis functions that you can apply to an image.

image source	The original input image.
imaging	Any process of acquiring and displaying images and analyzing image data.
inner gradient	Finds the inner boundary of objects.
inspection	The process by which parts are tested for simple defects, such as missing parts or cracks on part surfaces.
inspection function	Analyzes groups of pixels within an image and returns information about the size, shape, position, and pixel connectivity. Typical applications include testing the quality of parts, analyzing defects, locating objects, and sorting objects.
instrument driver	A set of high-level software functions, such as NI-IMAQ, that control specific plug-in computer boards. Instrument drivers are available in several forms, ranging from a function callable in a programming language to a VI in LabVIEW.
intensity	The sum of the red, green, and blue primary colors divided by three, (*Red* + *Green* + *Blue*)/3.
intensity calibration	Assigns user-defined quantities, such as optical densities or concentrations, to the gray-level values in an image.
intensity profile	The gray-level distribution of the pixels along an ROI in an image.
intensity range	Defines the range of gray-level values in an object of an image.
intensity threshold	Characterizes an object based on the range of gray-level values in the object. If the intensity range of the object falls within the user-specified range, it is considered an object. Otherwise it is considered part of the background.

J

jitter	The maximum amount of time that the execution of an algorithm varies from one execution to the next.
JPEG	Joint Photographic Experts Group. An image file format for storing 8-bit and color images with lossy compression. JPEG images have the file extension *JPG*.

JPEG2000	An Image file format for storing 8-bit, 16-bit, or color images with either lossy or lossless compression. JPEG2000 images have the file extension *JP2*.

K

kernel	A structure that represents a pixel and its relationship to its neighbors. The relationship is specified by the weighted coefficients of each neighbor.

L

labeling	A morphology operation that identifies each object in a binary image and assigns a unique pixel value to all the pixels in an object. This process is useful for identifying the number of objects in the image and giving each object a unique pixel intensity.
LabVIEW	Laboratory Virtual Instrument Engineering Workbench. A program development environment application based on the programming language G used commonly for test and measurement applications.
line gauge	Measures the distance between selected edges with high-precision subpixel accuracy along a line in an image. For example, this function can be used to measure distances between points and edges. This function also can step and repeat its measurements across the image.
line profile	Represents the gray-level distribution along a line of pixels in an image.
linear filter	A special algorithm that calculates the value of a pixel based on its own pixel value as well as the pixel values of its neighbors. The sum of this calculation is divided by the sum of the elements in the matrix to obtain a new pixel value.
local threshold	Creates a binary image by segmenting a grayscale image into a particle region and a background region.
logarithmic function	Increases the brightness and contrast in dark regions of an image and decreases the contrast in bright regions of the image.
logic operators	The image operations AND, NAND, OR, XOR, NOR, XNOR, difference, mask, mean, max, and min.

lossless compression	Compression in which the decompressed image is identical to the original image.
lossy compression	Compression in which the decompressed image is visually similar but not identical to the original image.
lowpass attenuation	Applies a linear attenuation to the frequencies in an image, with no attenuation at the lowest frequency and full attenuation at the highest frequency.
lowpass FFT filter	Removes or attenuates high frequencies present in the FFT domain of an image.
lowpass filter	Attenuates intensity variations in an image. You can use these filters to smooth an image by eliminating fine details and blurring edges.
lowpass frequency filter	Attenuates high frequencies present in the frequency domain of the image. A lowpass frequency filter suppresses information related to fast variations of light intensities in the spatial image.
lowpass truncation	Removes all frequency information above a certain frequency.
L-skeleton function	Uses an L-shaped structuring element in the skeleton function.
luma	The brightness information in the video picture. The luma signal amplitude varies in proportion to the brightness of the video signal and corresponds exactly to the monochrome picture.
luminance	*See* luma.
LUT	Lookup table. A table containing values used to transform the gray-level values of an image. For each gray-level value in the image, the corresponding new value is obtained from the lookup table.

M

M	(1) Mega, the standard metric prefix for 1 million or 10^6, when used with units of measure such as volts and hertz (2) Mega, the prefix for 1,048,576, or 220, when used with B to quantify data or computer memory.
machine vision	An automated application that performs a set of visual inspection tasks.

mask FFT filter	Removes frequencies contained in a mask (range) specified by the user.
match score	A number ranging from 0 to 1000 that indicates how closely an acquired image matches the template image. A match score of 1000 indicates a perfect match. A match score of 0 indicates no match.
MB	Megabyte of memory.
median filter	A lowpass filter that assigns to each pixel the median value of its neighbors. This filter effectively removes isolated pixels without blurring the contours of objects.
memory buffer	*See* buffer.
MMX	Multimedia Extensions. An Intel chip-based technology that allows parallel operations on integers, which results in accelerated processing of 8-bit images.
morphological transformations	Extract and alter the structure of objects in an image. You can use these transformations for expanding (dilating) or reducing (eroding) objects, filling holes, closing inclusions, or smoothing borders. They are used primarily to delineate objects and prepare them for quantitative inspection analysis.
M-skeleton function	Uses an M-shaped structuring element in the skeleton function.
multiple template matching	The technique used to simultaneously locate multiple grayscale templates within a grayscale image.

N

neighbor	A pixel whose value affects the value of a nearby pixel when an image is processed. The neighbors of a pixel are usually defined by a kernel or a structuring element.
neighborhood operations	Operations on a point in an image that take into consideration the values of the pixels neighboring that point.
NI-IMAQ	The driver software for National Instruments image acquisition hardware.
nonlinear filter	Replaces each pixel value with a nonlinear function of its surrounding pixels.
nonlinear gradient filter	A highpass edge-extraction filter that favors vertical edges.

nonlinear Prewitt filter	A highpass, edge-extraction filter based on 2D gradient information.
nonlinear Sobel filter	A highpass, edge-extraction filter based on 2D gradient information. The filter has a smoothing effect that reduces noise enhancements caused by gradient operators.
Nth order filter	Filters an image using a nonlinear filter. This filter orders (or classifies) the pixel values surrounding the pixel being processed. The pixel being processed is set to the *N*th pixel value, where *N* is the order of the filter.
number of planes (in an image)	The number of arrays of pixels that compose the image. A gray-level or pseudo-color image is composed of one plane. An RGB image is composed of three planes: one for the red component, one for the blue component, and one for the green component.

O

occlusion invariant matching	A geometric matching technique in which the reference pattern can be partially obscured in the target image.
OCR	Optical Character Recognition. The process of analyzing an image to detect and recognize characters/text in the image.
OCV	Optical Character Verification. A machine vision application that inspects the quality of printed characters.
offset	The coordinate position in an image where you want to place the origin of another image. Setting an offset is useful when performing mask operations.
opening	An erosion followed by a dilation. An opening removes small objects and smooths boundaries of objects in the image.
operators	Allow masking, combination, and comparison of images. You can use arithmetic and logic operators in NI Vision.
optical representation	Contains the low-frequency information at the center and the high-frequency information at the corners of an FFT-transformed image.
outer gradient	Finds the outer boundary of objects.

P

palette	The gradation of colors used to display an image on screen, usually defined by a CLUT.
particle	A connected region or grouping of pixels in an image in which all pixels have the same intensity level.
particle analysis	A series of processing operations and analysis functions that produce information about the particles in an image.
pattern matching	The technique used to quickly locate a grayscale template within a grayscale image
picture element	An element of a digital image. Also called pixel.
pixel	Picture element. The smallest division that makes up the video scan line. For display on a computer monitor, a pixel's optimum dimension is square (aspect ratio of 1:1, or the width equal to the height).
pixel aspect ratio	The ratio between the physical horizontal and vertical sizes of the region covered by the pixel. An acquired pixel should optimally be square, thus the optimal value is 1.0, but typically it falls between 0.95 and 1.05, depending on camera quality.
pixel calibration	Directly calibrates the physical dimensions of a pixel in an image.
pixel depth	The number of bits used to represent the gray level of a pixel.
PNG	Portable Network Graphic. An image file format for storing 8-bit, 16-bit, and color images with lossless compression. PNG images have the file extension *PNG*.
Prewitt filter	An edge detection algorithm that extracts the contours in gray-level values using a 3×3 filter kernel.
proper-closing	A finite combination of successive closing and opening operations that you can use to fill small holes and smooth the boundaries of objects.
proper-opening	A finite combination of successive opening and closing operations that you can use to remove small particles and smooth the boundaries of objects.

Q

quantitative analysis — Obtaining various measurements of objects in an image.

R

real time — A property of an event or system in which data is processed as it is acquired instead of being accumulated and processed at a later time.

resolution — The number of rows and columns of pixels. An image composed of *m* rows and *n* columns has a resolution of $m \times n$.

reverse function — Inverts the pixel values in an image, producing a photometric negative of the image.

RGB — A color encoding scheme using red, green, and blue (RGB) color information where each pixel in the color image is encoded using 32 bits: eight bits for red, eight bits for green, eight bits for blue, and eight bits for the alpha value (unused).

RGB U64 — A color encoding scheme using red, green, and blue (RGB) color information where each pixel in the color image is encoded using 64 bits:16 bits for red, 16 bits for green, 16 bits for blue, and 16 bits for the alpha value (unused).

Roberts filter — An edge detection algorithm that extracts the contours in gray level, favoring diagonal edges.

ROI — Region of interest.

(1) An area of the image that is graphically selected from a window displaying the image. This area can be used focus further processing.

(2) A hardware-programmable rectangular portion of the acquisition window.

ROI tools — A collection of tools that enable you to select a region of interest from an image. These tools let you select points, lines, annuli, polygons, rectangles, rotated rectangles, ovals, and freehand open and closed contours.

rotational shift	The amount by which one image is rotated relative to a reference image. This rotation is computed relative to the center of the image.
rotation-invariant matching	A pattern matching technique in which the reference pattern can be located at any orientation in the test image as well as rotated at any degree.

S

sample	An object in an image that you want to classify.
saturation	The amount of white added to a pure color. Saturation relates to the richness of a color. A saturation of zero corresponds to a pure color with no white added. Pink is a red with low saturation.
scale-invariant matching	A pattern matching technique in which the reference pattern can be any size in the test image.
segmentation function	Fully partitions a labeled binary image into non-overlapping segments, with each segment containing a unique particle.
separation function	Separates particles that touch each other by narrow isthmuses.
shape detection	Detects rectangles, lines, ellipses, and circles within images.
shift-invariant matching	A pattern matching technique in which the reference pattern can be located anywhere in the test image but cannot be rotated or scaled.
skeleton function	Applies a succession of thinning operations to an object until its width becomes one pixel.
smoothing filter	Blurs an image by attenuating variations of light intensity in the neighborhood of a pixel.
Sobel filter	An edge detection algorithm that extracts the contours in gray-level values using a 3×3 filter kernel.
spatial calibration	Assigns physical dimensions to the area of a pixel in an image.
spatial filters	Alter the intensity of a pixel relative to variations in intensities of its neighboring pixels. You can use these filters for edge detection, image enhancement, noise reduction, smoothing, and so forth.
spatial resolution	The number of pixels in an image in terms of the number of rows and columns in the image.

square function	*See* exponential function.
square root function	*See* logarithmic function.
standard representation	Contains the low-frequency information at the corners and high-frequency information at the center of an FFT-transformed image.
structuring element	A binary mask used in most morphological operations. A structuring element is used to determine which neighboring pixels contribute in the operation.
subpixel analysis	Finds the location of the edge coordinates in terms of fractions of a pixel.

T

template	A color or pattern that you are trying to match in an image using the color matching or pattern matching functions. A template can be a region selected from an image or it can be an entire image.
threshold	Separates particles from the background by assigning all pixels with intensities within a specified range to the particle and the rest of the pixels to the background. In the resulting binary image, particles are represented with a pixel intensity of 255 and the background is set to 0.
threshold interval	Two parameters: the lower threshold gray-level value and the upper threshold gray-level value.
TIFF	Tagged Image File Format. An image format commonly used for encoding 8-bit, 16-bit, and color images. TIFF images have the file extension *TIF*.
time-bounded	Describes algorithms that are designed to support a lower and upper bound on execution time.
tools palette	A collection of tools that enable you to select regions of interest, zoom in and out, and change the image palette.

V

value	The grayscale intensity of a color pixel computed as the average of the maximum and minimum red, green, and blue values of that pixel.
VI	Virtual Instrument. (1) A combination of hardware and/or software elements, typically used with a PC, that has the functionality of a classic stand-alone instrument. (2) A LabVIEW software module, which consists of a front panel user interface and a block diagram program.

W

Watershed Transform	Partitions an image based on the topographic surface of the image. The image is separated into non-overlapping segments with each segment containing a unique particle.

Index

Numerics

16-bit color images, 3-11
1D barcodes, reading, 5-35
8-bit color images, 3-11

A

acquiring measurement-ready images. *See* measurement-ready images, acquiring
analytic geometry measurements, 5-32
analyzing images, 2-12
application development
    general steps (figure), 1-7
    inspection steps (figure), 1-8
attaching calibration information to images, 2-12, 6-10
attenuation
    highpass, 2-17
    lowpass, 2-17
AVI files, 2-8

B

background correction algorithm, 4-2
barcodes
    reading data matrix barcodes, 5-35
    reading PDF417 barcodes, 5-35
binary images. *See* particle analysis
binary morphology, 4-3, 4-4

C

calibration
    attaching calibration information to images, 2-12, 6-10
    defining reference coordinate system, 6-3
    defining template, 6-2
    for perspective and nonlinear distortion, 6-1
    learning calibration information
        choosing learning algorithm, 6-6
        choosing ROI, 6-6
        correction table, 6-8
        learning score and error map, 6-7
        scaling mode, 6-8
        voiding calibration, 6-9
    overview, 2-2
    saving calibration information, 6-10
    simple calibration, 6-9
characters, reading, 5-34
circles, finding points along edge, 5-9
Classification Training Interface, 5-33
classifying samples, 5-33
color comparison, 3-12
color information, learning. *See* learning color information
color location algorithms for finding measurement points, 5-30
color measurements. *See* grayscale and color measurements
color pattern matching
    See also pattern matching
    defining search area, 5-26
    defining template images, 5-24
    setting matching parameters and tolerances
        color score weight, 5-29
        color sensitivity, 5-28
        minimum contrast, 5-29
        rotation angle ranges, 5-29
        search strategy, 5-28
    testing search tool on test images, 5-29
    training pattern matching tool using reference pattern, 5-25

color statistics
color comparison, 3-12
learning color information, 3-13, 3-14
choosing color representation sensitivity, 3-15
ignoring learned colors, 3-16
specifying information to learn, 3-13
primary components of color images (figures), 3-11
connectivity, 4-3
connector pane examples
image analysis, 2-4
image mask, 2-4
contour, finding points along edge, 5-11
control palette
Image Display control, 1-1, 2-9
IMAQ Image control, 1-1
Machine Vision controls, 1-2
NI Vision controls, 1-1
conventions used in the manual, *xi*
converting pixel coordinates to real-world coordinates, 5-30
convolution filters, 2-15
coordinate reference
building for machine vision
choosing method (figure), 5-7
edge detection, 5-3
pattern matching, 5-6
defining for calibration, 6-3
coordinates, converting pixel to real-world coordinates, 5-30
correction table, for calibration, 6-8
creating applications. *See* application development
creating images. *See* images

D

data matrix barcodes, reading, 5-35
deployment, application, *xiii*, A-9
determinism, A-5
diagnostic tools (NI resources), B-1
displaying
images, 2-8
results of inspection process, 5-37
distance measurements, 5-31
distortion, correcting. *See* calibration
documentation
conventions used in manual, *xi*
NI resources, B-1
related documentation, *xii*
drivers (NI resources), B-1
drivers, NI-IMAQ, 1-3, 2-2, A-1, A-2

E

edge detection
building coordinate reference, 5-3
finding measurement points
along multiple search contours, 5-12
along one search contour, 5-11
lines or circles, 5-9
error map, for calibration, 6-7
examples (NI resources), B-1
external window, displaying images, 2-8

F

Fast Fourier Transform (FFT), 2-16
filters
convolution, 2-15
highpass, 2-15
highpass frequency filters, 2-17
improving images, 2-14
lowpass, 2-15
lowpass frequency filters, 2-17
Nth order, 2-15
finding measurement points. *See* measurement points, finding

frequency domain, 2-17
function palettes
    Image Processing, 1-4
    Machine Vision, 1-5
    Vision Utilities, 1-2

G

geometric matching
    finding points, 5-18
    match mode, 5-21
    occlusion ranges, 5-22
    rotation angle ranges, 5-22
    scale factor ranges, 5-22
    training, 5-20
geometrical measurements, 5-32
golden template comparison, 5-36
grayscale and color measurements
    color statistics
        color comparison, 3-12
        learning color information, 3-12
        primary components of color images (figures), 3-11
    defining regions of interest
        interactively, 3-3
        programmatically, 3-8
        using masks, 3-9
    grayscale statistics, 3-10
        area, 3-10
        energy center, 3-10
        light intensity, 3-10
        maximum intensity, 3-10
        mean intensity, 3-10
        minimum intensity, 3-10
        percent of image analyzed, 3-10
        standard deviation, 3-10
grayscale morphology, 2-15

H

help, technical support, B-1
highpass filters, 2-15
highpass frequency filters
    attenuation, 2-17
    truncation, 2-17

I

ignoring learned colors, 3-16
image analysis, connector pane example, 2-4
Image Display control, 1-1, 2-9
image mask, connector pane example, 2-4
Image Processing function palettes, 1-4
images
    See also particle analysis
    acquiring or reading, 2-6
        figure, 1-7
    analyzing, 2-12
    attaching calibration information, 2-12, 6-10
    creating
        connector panes, 2-4
        Image Dst input, 2-5
        Image Mask input, 2-4
        multiple images, 2-3
        overview, 2-2
        source images for destination image, 2-5
        valid image types, 2-2
    degradation, A-4
    displaying, 1-1, 2-8
    improving
        advanced operations, 2-18
        FFT (Fast Fourier Transform), 2-16
        filters, 2-15
        grayscale morphology, 2-15
        lookup tables, 2-14

imaging system
calibrating, 2-2
setting up, 2-1
IMAQ ArrayToComplexImage VI, 2-18
IMAQ ArrayToImage VI, 2-8
IMAQ AutoBThreshold VI, 4-2
IMAQ AVI Close VI, 2-8
IMAQ AVI Open VI, 2-8
IMAQ AVI Read Frame VI, 2-8
IMAQ Bisecting Line VI, 5-32
IMAQ Browser Delete VI, A-11
IMAQ Browser Focus Setup VI, A-11
IMAQ Browser Focus VI, A-11
IMAQ Browser Insert VI, A-11
IMAQ Browser Replace VI, A-11
IMAQ Browser Setup VI, A-11
IMAQ Centroid VI, 3-10
IMAQ Check Timed Execution VI, A-9
IMAQ Clamp Horizontal Max VI, 5-31, 5-38
IMAQ Clamp Horizontal Min VI, 5-31, 5-38
IMAQ Clamp Vertical Max VI, 5-31, 5-38
IMAQ Clamp Vertical Min VI, 5-31, 5-38
IMAQ Classify VI, 5-33
IMAQ Clear Overlay VI, 5-38
IMAQ Close VI, 2-7
IMAQ ColorLearn VI, 3-12, 3-15
IMAQ ColorMatch VI, 3-12
IMAQ ColorThreshold VI, 4-3
IMAQ ColorToRGB VI, 3-12
IMAQ Compare Golden Template VI, 5-36
IMAQ Compare Golden Template, inspecting image for defects, 5-36
IMAQ ComplexAttenuate VI, 2-17
IMAQ ComplexImageToArray VI, 2-18
IMAQ ComplexPlaneToImage VI, 2-18
IMAQ ComplexTruncate VI, 2-17
IMAQ Concentric Rake VI, 5-12
IMAQ ConstructROI VI, 3-5
IMAQ Convert Annulus to ROI VI, 3-8
IMAQ Convert Line to ROI VI, 3-8
IMAQ Convert Pixel to Real World VI, 5-30, 5-36
IMAQ Convert Point to ROI VI, 3-8
IMAQ Convert Rectangle to ROI (Polygon) VI, 3-9
IMAQ Convert Rectangle to ROI VI, 3-8
IMAQ Convert ROI to Annulus VI, 3-9
IMAQ Convert ROI to Line VI, 3-9
IMAQ Convert ROI to Point VI, 3-9
IMAQ Convert ROI to Rectangle VI, 3-9
IMAQ Convolute VI, 2-15
IMAQ Count Object VI, 5-38
IMAQ Create VI, 2-2
IMAQ Danielsson VI, 4-4
IMAQ Dispose Classifier VI, 5-33
IMAQ Dispose VI (note), 2-3
IMAQ Draw Text VI, A-11
IMAQ Draw VI, A-11, A-12
IMAQ Edge Tool VI, 2-14, 5-11
IMAQ Equalize VI, 2-14
IMAQ ExtractColorPlanes VI, 3-10, 3-11
IMAQ ExtractSingleColorPlane VI, 3-11
IMAQ FFT VI, 2-16
IMAQ FillHoles VI, 4-5
IMAQ Find Circular Edge VI, 5-10, 5-38
IMAQ Find Concentric Edge VI, 5-9, 5-38
IMAQ Find CoordSys (2 Rects) VI, 5-3, 5-4, 5-38
IMAQ Find CoordSys (Pattern) 2 VI, 5-6
IMAQ Find CoordSys (Pattern) VI, 5-6, 5-38
IMAQ Find CoordSys (Rect) VI, 5-3, 5-38
IMAQ Find Horizontal Edge VI, 5-9, 5-37
IMAQ Find Pattern VI, 5-38
IMAQ Find Vertical Edge VI, 5-9, 5-37
IMAQ Fit Circle 2 VI, 5-32
IMAQ Fit Ellipse 2 VI, 5-32
IMAQ Fit Line VI, 5-32
IMAQ Get LCD ROI VI, 5-33

IMAQ Get Meter 2 VI, 5-32
IMAQ Get Meter VI, 5-32
IMAQ GetFileInfo VI, 2-7
IMAQ GetPalette VI, 2-9
IMAQ GetPointsOnLine VI, 5-12
IMAQ Grab Acquire VI, 2-7
IMAQ Grab Setup VI, 2-7
IMAQ GrayMorphology VI, 2-16
IMAQ Histogram VI, 2-13
IMAQ Histograph VI, 2-13
IMAQ Image control, 1-1
IMAQ Image To Image VI, A-12
IMAQ ImageToArray VI, 2-8
IMAQ ImageToComplexPlane VI, 2-18
IMAQ Initialize Timed Execution VI, A-6
IMAQ IntegerToColorValue VI, 3-12
IMAQ Inverse VI, 2-14
IMAQ Label VI, 3-10
IMAQ Learn Color Pattern VI, 5-26
IMAQ Learn Multiple Geometric Patterns VI, 5-23
IMAQ Learn Pattern 2 VI, 5-15
IMAQ Light Meter (Line) VI, 3-6, 3-10
IMAQ Light Meter (Point) VI, 3-6, 3-10
IMAQ Light Meter (Rectangle) VI, 3-6, 3-10
IMAQ LineProfile VI, 2-13
IMAQ Lines Intersection VI, 5-32
IMAQ Local Threshold VI, 4-2
IMAQ LowPass VI, 2-15
IMAQ Mask VI, 4-4
IMAQ Match Geometric Pattern VI, 5-19
IMAQ Match Multiple Geometric Patterns VI, 5-23
IMAQ MathLookup VI, 2-14
IMAQ Merge Overlay VI, A-12
IMAQ Mid Line VI, 5-32
IMAQ Morphology VI, 4-5
IMAQ MultiThreshold VI, 4-2
IMAQ OCR Dispose Session VI, 5-34, 5-36
IMAQ OCR Read Character Set File VI, 5-34, 5-36
IMAQ OCR Read Text VI, 5-34, 5-36
IMAQ Overlay Arc VI, 5-37
IMAQ Overlay Bitmap VI, 5-37
IMAQ Overlay Line VI, 5-37
IMAQ Overlay Multiple Lines VI, 5-37
IMAQ Overlay Oval VI, 5-37
IMAQ Overlay Points VI, 5-37
IMAQ Overlay Rectangle VI, 5-37, A-12
IMAQ Overlay ROI VI, 5-37
IMAQ Overlay Text VI, 5-37
IMAQ Particle Analysis Report VI, 4-5
IMAQ Particle Analysis VI, 4-5
IMAQ Particle Filter 2 VI, 4-4
IMAQ Perpendicular Line VI, 5-32
IMAQ Point Distance VI, 5-31
IMAQ Polygon Area VI, 5-32
IMAQ Quantify VI, 3-10
IMAQ Rake VI, 5-12
IMAQ Read Barcode VI, 5-35
IMAQ Read Classifier File VI, 5-33
IMAQ Read Data Matrix Barcode VI, 5-35
IMAQ Read Image and Vision Info VI, 2-7, 5-39
IMAQ Read LCD VI, 5-33
IMAQ Read Meter VI, 5-32
IMAQ Read Multiple Geometric Template VI, 5-23
IMAQ Read PDF417 Barcode VI, 5-35
IMAQ Read Single Digit VI, 5-33
IMAQ ReadFile VI, 2-7
IMAQ RejectBorder VI, 4-3
IMAQ Remote Display Options VI, A-3
IMAQ RemoveParticle VI, 4-3
IMAQ ReplaceColorPlanes VI, 3-10
IMAQ RGBToColor2 VI, 3-12
IMAQ ROIProfile VI, 2-13, 5-11
IMAQ ROIToMask VI, 3-5
IMAQ RT Video Out VI, A-4
IMAQ Select Line VI, 3-6

IMAQ Select Point VI, 3-6
IMAQ Select Rectangle VI, 3-6, 5-31
IMAQ Separation VI, 4-4
IMAQ Sequence VI, 2-7
IMAQ Setup Learn Color Pattern VI, 5-26
IMAQ Setup Learn Pattern 2 VI, 5-15
IMAQ Setup Match Color Pattern VI, 5-27
IMAQ Setup Match Pattern 2 VI, 5-17
IMAQ Simple Edge VI, 5-11
IMAQ Snap VI, 2-7
IMAQ Spoke VI, 5-12
IMAQ Start Timed Execution VI, A-8, A-9
IMAQ Stop Timed Execution VI, A-8
IMAQ Stop VI, 2-7
IMAQ Threshold VI, 4-1
IMAQ Uninitialize Timed Execution VI, A-9
IMAQ UserLookup VI, 2-14, 4-2
IMAQ Video Out Display Mode VI, A-4, A-13
IMAQ Vision Remote Server VI, A-10
IMAQ Watershed Transform VI, 4-4
IMAQ WindClose VI, 2-9
IMAQ WindDraw VI, 2-8, A-3, A-4, A-10
IMAQ WindLastEvent VI, 2-12
IMAQ WindMove VI, 2-8
IMAQ WindSetup VI, 2-8
IMAQ WindToolsClose VI, 3-4
IMAQ WindToolsMove VI, 3-4
IMAQ WindToolsSelect VI, 3-4
IMAQ WindToolsSetup VI, 3-4
IMAQ WindToolsShow VI, 3-4
IMAQ Write Image and Vision Info VI, 5-15, 5-26, 5-38
IMAQ Write Multiple Geometric Template VI, 5-23
inspecting image for defects, 5-36
Inspection
  Defects, 5-36
  Identify Parts Under, 5-33
  palette, 1-6
  Verify Characters, 5-36
instrument drivers, 1-3, 2-2, A-1, A-2
instrument drivers (NI resources), B-1
instrument reader measurements, 5-32

J

JPEG, 2-7
JPEG2000, 2-7

K

KnowledgeBase, B-1

L

LabVIEW Real-Time Module, *xiii*, A-1
learning calibration information
  choosing learning algorithm, 6-6
  choosing ROI, 6-6
  correction table, 6-8
  learning score and error map, 6-7
  scaling mode, 6-8
  voiding calibration, 6-9
learning color information
  choosing color representation sensitivity, 3-15
  entire image, 3-13
  ignoring learned colors, 3-16
  multiple regions in image, 3-14
  region in image, 3-14
  specifying information to learn, 3-13
lines, finding points along edge, 5-9
locating objects to inspect. *See* machine vision
lookup table transformations, 2-14
lowpass filters, 2-15
lowpass frequency filters
  attenuation, 2-17
  truncation, 2-17

M

machine vision
converting pixel coordinates to real-world coordinates, 5-30
defining region of interest for search area
interactively, 5-8
programmatically, 5-9
displaying results, 5-37
finding measurement points
color location, 5-30
color pattern matching, 5-24
edge detection, 5-9
pattern matching, 5-13
locating objects to inspect
choosing method for building coordinate reference (figure), 5-7
edge detection for building coordinate reference, 5-3
pattern matching for building coordinate reference, 5-6
making measurements
analytic geometry measurements, 5-32
distance measurements, 5-31
instrument reader measurements, 5-32
overview, 5-1
steps for performing (figure), 5-2
Machine Vision controls, 1-2
Machine Vision function palettes, 1-5
masks, for defining regions of interest, 3-9
measurement points, finding
color location, 5-30
color pattern matching, 5-24
edge detection, 5-9
pattern matching, 5-13
measurement-ready images, acquiring
acquiring or reading images, 2-6
analyzing images, 2-12
attaching calibration information, 2-12
calibrating imaging system, 2-2
creating images, 2-2
displaying images, 2-8
improving images
advanced operations, 2-18
FFT (Fast Fourier Transform), 2-16
filters, 2-15
grayscale morphology, 2-15
lookup tables, 2-14
setting up imaging system, 2-1
morphology
binary, 4-3
grayscale, 2-15
multiple template images, 5-23

N

National Instruments support and services, B-1
NI OCR Training Interface, 5-36
NI support and services, B-1
NI Vision Control, 1-1
NI Vision for LabVIEW
creating applications
general steps (figure), 1-7
inspection steps (figure), 1-8
function palettes
Image Processing, 1-4
Machine Vision, 1-5
Vision Utilities, 1-2
overview, 1-1
NI Vision Template Editor, 5-19
Niblack algorithm, 4-2
NI-IMAQ, *xiii*, 2-2, A-1, A-2
Nth order filters, 2-15

O

OCR, 5-34

P

particle analysis
connectivity, 4-3
creating binary image, 4-1
improving binary image
improving particle shapes, 4-5
removing unwanted particles, 4-3
separating touching particles, 4-4
particle measurements, 4-5
particle measurements, 4-5
pattern matching
See also color pattern matching
building coordinate reference, 5-6
finding measurement points
defining and creating template images, 5-13
defining search area, 5-15
general steps, 5-13
learning the template, 5-15
setting matching parameters and tolerances, 5-17
testing search tool on test images, 5-18
verifying results with ranking method, 5-18
PDF417 barcodes, reading, 5-35
perspective errors, calibrating. *See* calibration
pixel coordinates, converting to real-world coordinates, 5-30
points, finding. *See* measurement points, finding
programming examples (NI resources), B-1

R

ranking method for verifying pattern matching, 5-18
reading
1D barcodes, 5-35
AVI files, 2-8
barcodes, 5-35
characters, 5-34
data matrix barcodes, 5-35
PDF417 barcodes, 5-35
reading images, 2-6
regions of interest, defining
for calibration, 6-6
interactively
displaying tools palette in separate window, 3-4
for machine vision inspection, 5-8
ROI constructor window, 3-5
tools palette functions (table), 3-2
tools palette tools and information (figure), 3-7
programmatically
for machine vision inspection, 5-9
specifying ROI elements and parameters, 3-8
using VIs, 3-8
using masks, 3-9
related documentation, *xii*
remote display, A-3
resource management, A-5
ROI. *See* regions of interest, defining
RT Video Out, A-2, A-4, A-13

S

samples, classifying, 5-33
saving calibration information, 6-10
scaling mode, for calibration, 6-8
search algorithm
    testing, 5-22
search contour, finding points along edge, 5-11
software (NI resources), B-1
specifying information to learn
    using entire image, 3-13
    using multiple regions in image, 3-14
    using region in image, 3-14
support, technical, B-1

T

technical support, B-1
template for calibration, defining, 6-2
template images
    defining
        color pattern matching, 5-25
        pattern matching, 5-13
    defining and creating, 5-19
    learning
        color pattern matching, 5-25
        pattern matching, 5-15
    multiple, 5-23
time-bounded execution, A-4, A-5, A-6
tools palette functions (table), 3-2
training and certification (NI resources), B-1
troubleshooting (NI resources), B-1
truncation
    highpass, 2-17
    lowpass, 2-17

V

validating calibration, 6-9
verifying pattern matching, 5-18
Vision Utilities function palettes, 1-2

W

watershed transform, 4-4
Web resources, B-1